Chemical Markup, XML, and the World Wide Web. 6. CMLReact, an XML Vocabulary for Chemical Reactions

Gemma L. Holliday,[†] Peter Murray-Rust,[†] and Henry S. Rzepa*,[‡]

Department of Chemistry, Unilever Center for Molecular Informatics, Lensfield Road,
Cambridge CB2 1EW, U.K., and Department of Chemistry, Imperial College London,
South Kensington Campus, London SW7 2AZ, U.K.

Received July 1, 2005

A set of components (CMLReact) for managing chemical and biochemical reactions has been added to CML. These can be combined to support most of the strategies for the formal representation of reactions. The elements, attributes, and types are formally defined as XMLSchema components, and their semantics are developed. New syntax and semantics in CML are reported and illustrated with 10 examples.

1. INTRODUCTION

CML (Chemical Markup Language) has been developed[1] as an XML application for the representation of compounds, molecules and molecular computations, crystallography, and spectra. In this article, we introduce CMLReact, the application of CML, to the important area of chemical reactions. The overall structure of the article is

1. This introduction
2. Review of representation of chemical reactions
   a. Strategies for Reaction Formalization
   b. Current Practice in Representation of Reactions
3. Strategies for representation of reactions
   a. Design of CMLReact
   b. Vocabulary for CMLReact and the XMLSchema
   c. The New Elements and attributes in CML
4. Examples of CMLReact
5. Creating and Processing CMLReact

CML has now evolved into a modular architecture of XML Schema (XSD) components from which applications can be constructed. In its use of XMLSchema, CML specifies the vocabulary (element names, attribute names), data types (mainly character data content of attributes and elements), and in some cases the enumeration of allowed values. It does not generally use tightly controlled content models for element content, as this restricts flexibility of representation. In a recent article[2] we outlined the principles of this design, including concepts such as

- The component-based approach
- Validation (XSD and other)
- Schema-based addition of semantics
- Custom implementation of semantics
- The creation of valid instances
- Software to analyze and process instances
- Extension or restriction of CML functionality

Each component is designed to be as context-free as possible (i.e. its interpretation does not depend on siblings, ancestors, or dependents). Where this can be achieved it is possible to create per-component tools and processes. It is also possible to mix-and-match components without breaking software, analogous to construction toys (Meccano(TM), Lego(TM), TinkerToy(TM), etc.). "CMLReact" therefore describes a set of components which adds chemical reactions to current CML functionality. CMLReact does not have a separate namespace from other CML components; instead this is viewed as a major release of CML which be used with **http://www.xml-cml.org/schema/cml3** as a namespace. In what follows below, monospaced fonts (e.g. **reaction**) are used to denote components of the CML vocabulary. Their general meaning should be clear, prior to their specification later in the article.

The following principles are useful while reading this article:

- The vocabulary and some enumerated content is inflexible and can be validated. However XSD is too weak for validating chemistry. For example an **atom** cannot be mapped to a **bond** during a reaction, but an **electron** child of an atom can be **map**ped to an **electron** child of a **bond**.
- The models for element content are variable and can be customized to reflect different aspects of chemistry. It might, for example, make sense for a **reactionScheme** to contain unique **identifier**s, **spectrum**s to record progress, **math:*** to record kinetics, or **svg:*** to contain rendered **reactionScheme**s.
- The semantics of components are not (yet) enforceable by specific language specifications although we are developing XSLT (extensible stylesheet language transforms) and RDF (Resource description framework) for this. CML has been developed with the concept of an implicit communal "mainstream" of the representation of chemistry, including reactions. It is a major task to formalize this and until then we rely on

- a wide variety of examples

- a reference toolkit showing (at least) our core interpretation of chemistry

- human-understandable description. Thus **mechanism** has semantics which should honor the IUPAC definition (see below) even though the element content is unspecified.

* Corresponding author e-mail: h.rzepa@ic.ac.uk.
[†] Unilever Center for Molecular Informatics.
[‡] Imperial College London.

10.1021/ci0502698 CCC: $33.50 © 2006 American Chemical Society
Published on Web 11/30/2005

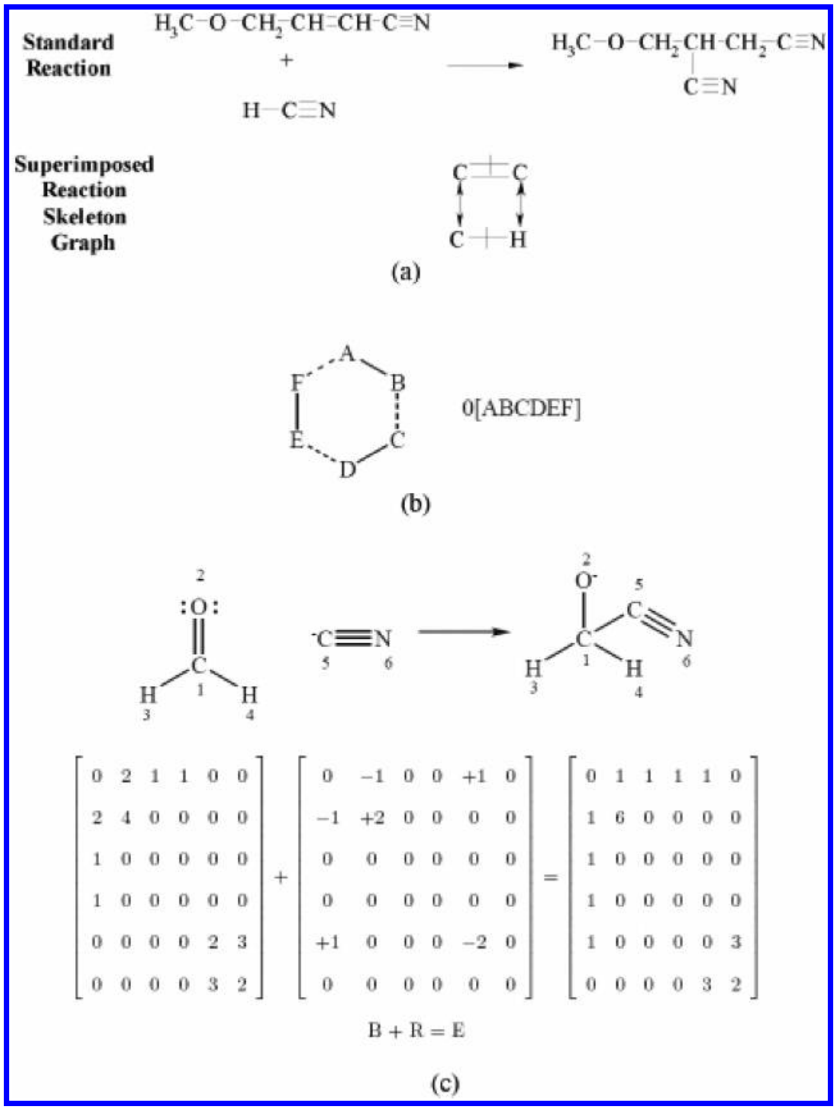


Figure 1. Examples of reaction depiction (a) Valdutz representation, (b) Hendrickson's representation, and (c) the Dugundji−Ugi matrix representation.

We ask that developers honor the spirit of this and, where semantics are unclear, raise the problems in public areas such as the CML-discuss list.[3]

2. REPRESENTATION OF CHEMICAL REACTIONS

Compared to molecules and compounds, reactions are much less freely represented and interchanged in electronic form. This is partly due to the different motivations for formalizing a reaction, and therefore the greater difficulty in creating a generic specification. In some cases extensions of molecular formats have been used (e.g. RXN, SMILES, SMIRKS, CDX), while in others a matrix or graph formalization has been developed. Often the representation is tied to a specific piece of software (e.g. for developing synthetic strategy such as LHASA). In yet others the reaction is managed by a relational database (e.g. KEGG) where the products and reactions are linked to other tables. Moreover almost all software for manipulating reactions has been developed by commercial or quasi-commercial organizations. Historically these have not promoted interoperability, and in most cases the author or user of reaction data has to buy software from a supplier. XML is not in common use here.

2.1. Strategies for Reaction Formalization. Processing of reactions requires information on the reaction center, bonds cleaved and formed, and how valence electrons are rearranged during the course of a reaction. A chemical reaction is characterized by its reactants, products, and reaction conditions. These conditions include reagents, catalysts, light levels, temperature, pressure, etc. A reaction mechanism is described with the "curly arrows" over a linked chain or cycle of atoms. The aggregate of atoms that exchange bonds is called the reaction center, and "a unit reaction" is an attempt to describe the irreducible representation of bond and electronic change. More complex transformations may then be analyzed as a sequence of unit reactions.

The following literature has guided the design of CML-React:

- Ingold's systematic representation of reactions in the 1930s (S_N2, E1, "curly arrows", etc.) [4]

- Classifications of reaction types (e.g. Theilheimer's addition, elimination, rearrangement and substitution (exchange).[5]

- Notations for serialization and graphical representation. Some of these are primarily lexical (e.g. serialization) but others describe conceptual abstraction of the reaction center (Figure 1).[6]

- Algorithms for formalizing the changes in a reaction.[7]

- Database searching. Some authors (e.g. ref 7a) have created a model driven classification system designed to afford rapid retrieval of reactions from a database.• Graphical or topological approaches, where in principle the graph can be converted into a labeled connection table. There is no agreed formalism for bonds, which may be fractional (e.g. "0.5", "1.5"), stylistic ("dashed", "dotted"), or with annotations for which are made and broken (Figure 1b).

- Matrix representation (ref 7e). Figure 1c shows that B(reactant) and E(product) represent implicitly labeled atoms, the two-center bond order between them, and the free valence electrons on each. The **R** matrix is the change during the reaction. For a multistep reaction with all atoms and invariant labeling the **E** matrix of one step(1) can become the **B** matrix of another (step2):

$$\mathbf{B}1 + \mathbf{R}1 => \mathbf{E}1\ (=\mathbf{B}2) + \mathbf{R}2 => \mathbf{E}2$$

CML implicitly represents such matrices, which have great power such as for animating enzyme reaction mechanisms.

The following is an illustrative (but not exhaustive) list of information or knowledge about a given system that would be needed for formalizing its representation as a chemical reaction.

- the precise description of the trajectory of atomic ensembles on the potential energy surface, optionally with excited states

- the identification of two local minima and an intervening transition state

- the identities of reactant species and product species

- multiple successive or alternative reaction steps within a "reaction". This often involves formal electron-counting (e.g. curly arrows).

- the overall stoichiometry of a process, perhaps with associated physical data (heats, rates, etc.)

- the procedures required to carry out a reaction in a laboratory with the results and observations obtained

- a reaction type (e.g. esterification) where generic groups replace specific atoms

- a reaction scheme, whereby a set of reactions is laid out to show an overall synthetic strategy or enzymatic cascade

- atom and bond mapping (e.g. where the identity of reactant and product atoms is known as in ester hydrolysis)

- computation of a reaction—Each atom retains its identity throughout the process, and no mapping is required. A bond is then completely identified by the atoms it connects.

2.2. Current Practice in Representation of Reactions. Summarized below are the approaches used by some of the major program systems in common use for authoring, storing, and querying reactions. We emphasize that this is not a comprehensive list, as we do not have open access to some systems, and others are used in niche chemical disciplines. The following protocols and tools are widely used, and for the most part well documented, although the documentation is often directed at humans; machine-interpretation may be less clear, as for example the specification of a "dotted bond". We have little evidence that reactions can be interchanged usefully between these systems.

- CambridgeSoft *ChemDraw*. A graphically oriented program where reactions are denoted by one or more arrows between multiple molecules. Components can be visually positioned without semantic restraints. The CDX and CDXML formats provide support for reactions through the **step** concept which labels **fragment**s as **reactionStepProduct**s or **reactionStepReactant**s. The identification of a product or reactant is determined by its precise positioning on the diagram and the use of certain arrow glyphs.
- MDL *ISIS/Draw*. The RXN file format has product and reactant molecules and an atom−atom mapping facility. However it is also possible to generate RXN files where the separate reactants are combined into a multimolecule without separate components.
- Daylight *Reaction SMILES* is a superset of SMILES used for expressing reactions in the general form: reactants > agents > products, e.g:

CCO.CC(=O)O>[H+]>CC(=O)OCC.O

Although this does provide atom−atom mapping, it is less easily extensible.

3. STRATEGIES FOR REPRESENTATION OF REACTIONS

We first emphasize that chemists use different dimensionalities in describing molecules and compounds, and the following categories[8] may be a useful (albeit approximate) classification:

- "0D". The identification of a compound, perhaps through a name or registry number, and often with a compositional formula (e.g. C7H6O).
- "1D". The connectivity of atoms ("structural formula"), possibly including atom and bond-based stereochemistry. This forms the basis of the IUPAC InChI identifier.[9]
- "2D". x- and y- coordinates are used to lay out the atoms on "paper"—the minimum requirement for "curly arrows". The coordinates have no formal relation to "3D" coordinates. A serious problem is that the graphical placement of species and the precise glyphs used to annotate them is often critical in understanding their semantics. Thus placing a compound over a right-arrow is an indication that it has a role as a reactant, catalyst, solvent, or spectator (examples are discussed in ref 7).
- "3D". x-, y- z- coordinates represent the coordinates of the atomic nuclei in Cartesian space. They are widely used to compute properties of reactions. For enzyme reactions the experimental structures of proteins (from crystallography or NMR) with substrates, cofactors, and analogues are often excellent starting points for deducing the changes in 3D coordinates during the reaction(s). For solid-state reactions 3D coordinates may be the only means of representing the precise details of the reaction.

CML is able to hold all of these independently for a given molecule, and this is maintained for components with CMLReact. We note that there are many other annotations and attributes already developed for CML and that these and extensions are available for CMLReact. They include

- observed, calculated, or assumed properties of components (molecules, atoms, bonds, electrons)
- labels and identifiers
- mappings

We have found the following conceptual hierarchy of representation to be useful:

1. A “reaction scheme”, consisting of several reactions, with a wide variety of topologies (cycles, branches, parallels, etc.). Common examples are biochemical pathways (e.g. tricarboxylic acid cycle) or directed synthesis (e.g. of an organic molecule).

2. Identification of a reaction, but without details of molecules or atoms involved. This has been common for biochemical pathways.

3. Enumeration of the reactants and products in a reaction, but without formulas or stoichiometry.

4. Enumeration of the reactants and products in a reaction, with formulas and optional stoichiometry.

5. Deconstruction of a single “reaction” into a series of mechanistic “steps”. These can be based on physical measurements (e.g. to identify transient species), bio/chemical probes, or computation and are often inferred by analogy. The distinction between a “reaction” and a “reaction step” may depend on the discipline and motivation of the scientist.

6. Representation of a reaction step with changes in the positions of atomic nuclei, the presence and order of formal bonds and the formalized “position” of electrons for bookkeeping (“lone pairs”, “pi-electrons”, “octet”, etc.). If a concept can be held in CML, then changes in it can form the basis of a reaction in CMLReact. Thus the excitation of a molecule from a singlet to a triplet or the transfer of an electron from one location to another could be represented as a step. Nuclear reactions could be described using the **isotopeNumber** attribute on **atom**.

3.1. Design of CMLReact. Where possible we have used terminology consistent with the IUPAC Gold Book.[10]

- The CMLReact vocabulary carries only the general connotation of “transformation” and no default implication of conservation laws. Thus the following processes could be represented:

1. lead → gold

2. glucose → ATP

3. boat → chair

4. light → glucose

5. $MnO_4^- + 8H^+ + 5e^- \rightarrow Mn^{2+} + 4H_2O$

Only the last of these conserves mass and charge in the representation.

- The design provides about 10 components (enumerated below) which can be combined in a variety of groupings. This allows flexibility in the interpretation of concepts such as **reactionStep**. Because of the wide variation in reactions the hardcoded semantics in CMLReact are normally associated only with the components and not their context. Thus a **molecule** does not contain its role in a **reaction** or the **count** of its stoichiometry, which are managed by the containing **reactant**. For interpreting context a higher layer, such as RDF, is proposed (but not described in detail in this article).

- The semantics of the components are limited to

1. calculation of the atomic components and charges in XMLElements

2. annotation of any component through its ID

3. identification of products, reactants, other reaction components, mechanisms and, properties

4. provision of **role** and **scheme** attributes for semicontrolled vocabularies

5. author’s assertion (**type**) of the directionality of reaction

6. **count, yield**, and **ratio** which allow mass balance to be computed

- The semantics of the groupings are limited to

1. related “reactions” within **reactionScheme** or **reactionStepList**. A few common relations are defined (e.g. **consecutive**, **simultaneous**).

2. the ability to decompose such a group into primitive steps or to compose steps into a group. This is possible through the use of IDs on all elements so that semantic processors can, for example, identify common explicit or implicit hierarchies.

3. fairly general content models. It is currently too restrictive to require precise specification of child elements for (say) **reactionScheme**. We believe that semantic constraints are now better imposed by RDF and similar annotations rather than XMLSchema content models.

- Although “curly arrows” can be incorporated into CMLReact at this stage we have decided to omit details from this description. Although an arrow apparently represents electron movement, it is almost always a mapping between electrons in reactants and products. A suggested scheme is given in example 1 below. We note that the components (**atom**, **electron**, **bond**, and also **map** and **link**) already exist in CML so that only the semantics need formalization.

- The Ugi approach can be easily derived from the **atom**, **bond**, and **map** elements if all the equivalences are known.

- CMLReact objects can be stored and queried by normal XML methods (e.g. XPath), but the language does not currently formalize the representation of reaction queries.

Our requirements are therefore

- That a machine can validate the reaction by analyzing the components and certain relations between them (mainly stoichiometry). (Most current tools do not by default validate stoichiometry).

- That a machine be able to understand the reaction independently of the program that created it.

- That the reaction should be storable in a database or other repository and should be searchable by chemical concepts (e.g. change in formal bond orders).

- That one or more reactions or series of steps be analyzable by machine. Major motivations are the machine-classification of reactions, and tools for “predicting reactions”. This includes synthesis planning and metabolic flux.

- That the precise components of the reaction can be fed into computational codes such as MO or similar approaches.

- That the reaction can be rendered (displayed) in different ways.

- That components can be annotated through mechanisms defined by other XML technologies such as RDF.

- That several reactions can be combined into a reaction scheme or a scheme decomposed into reactions.

- That human annotations and classifications can be easily identified

We note that most components defined in a chemical XMLSchema will require procedural programming support. Thus a **reaction** might require code to calculate the stoichiometric changes and perhaps to interpret some of the commoner properties. The components in CMLReact, therefore, are designed as much as possible to support procedural, functional, and ontological programming. Moreover they have to be as context-free as possible—the code associated with a reactant should ideally be independent of the context in which the element could be found.

3.2. CMLReact Vocabulary. All components are indefinitely extensible and can hold any allowed CML attributes and/or children or additional namespaced components outside CML. CML originally contained the following elements to deal with reactions: **reaction** and **reactionList**; **reactant** and **reactantList**; **product** and **productList**; **substance** and **substanceList**; and **property** and **propertyList**.

These are sufficient for most single-step reactions but require enhancement for reaction schemes and mechanists information. The CMLReact schema now includes XML constructions for

- reactant, product, spectator (e.g. protein side chains), or substance (which can include solvent, catalysts, etc.)
- mechanism and transition state
- annotations (e.g. titles, names and labels)
- properties, both controlling conditions and observations/measurements
- atoms, bonds, and electrons. These can be explicitly included with a range of attributes (position, annotation, displayHints, etc.).

We also find the following to be frequently required:

- **reactant** (**reactantList**) and **product** (**productList**) which includes reactant and product **count**s
- **reactiveCentre**, which includes **bondType**, **atomType**, **bondSet**, and **atomSet**. Synonyms or alternative spellings for this element (**reaction center**, **reacting center**, and **reacting center** are all also in common use) can be handled by including a dictionary reference.
- **mechanism** and **mechanismComponents**
- **spectator** and **spectatorList**
- **substance** and **substanceList**
- **property** and **propertyList**
- Reaction step(s) (**reactionScheme**, **reactionStepList**, **reactionStep**, and **reaction**).

3.3. The New Elements and Attributes in CML. As before, all CML elements (old and new) can have **id** and **ref** attributes. **reactionScheme** and **reactionStepList** add power to the original element **reactionList** in CMLSchema, which had no semantics, being a simple container. **reactionScheme** and **reactionStepList** can contain a variety of elements, including nesting, and support a wide range of reaction topologies (see examples).

3.3.1. reactionScheme. This is designed to contain a set of reactions which are linked in some way but without hardcoded semantics. It is often represented graphically by a reaction topology that can include branches and cycles. Typical examples are

- biochemical pathways
- multistep chemical syntheses, often with convergent or parallel components
- radical chain reactions.

The components in a **reactionScheme** may be primitive reactions or more complex nested **reactionScheme**s or **reactionStepList**s. The semantic relations between components of a **reactionScheme** are heterogeneous and may not be describable by XML containment. (This is a difference from **reactionStepList**.) We believe however that RDF annotation will be a powerful way of adding these semantics.

The design was kept general as the term *reaction* can be used in several ways. For example a biochemical pathway involves many coupled enzyme reactions, each of which is a reaction in its own right as the end point of the reaction is at a local minima on the reaction pathway. Each of these reactions are complex within themselves and can include several **mechanistic** steps, that are, again, reactions themselves. Thus **reactionScheme** can include both **reaction**s and **reactionStepList**s that provide a more detailed view of the individual components. The **reactionScheme** describes a complex of reactions with metadata, one (or more) overall **reaction**s, and a **reactionStepList**.

The element **reactionStep** contains either a **reaction** or **reactionScheme** and can carry information on yields and ratios.

3.3.2. reactionStep and reactionStepList. This describes a series of **reactionStep**s. It must always be contained within a **reactionScheme** as it is designed to manage “subreactions” that have close relationships. These subreactions will often, when taken together, describe a finer grained level of reaction or reaction type. These normally have a close relationship, the most common being **consecutive** and **simultaneous**. Each **reactionStep** will normally contain a single **reaction**, though it is allowable to include nested **reactionScheme**s (but not directly **reactionStepList**s) when finer detail is required. They are most likely to describe elucidated steps in a reaction mechanism. Steps in a synthesis are probably better described by a scheme.

Reversible reactions are identified by a **type** attribute on the **reaction**. This attribute simply states that the author has described the **reaction** as reversible or irreversible. The attribute implies no deeper chemical semantics (it can be argued that all reactions are reversible). However it allows a search for a **reactant** to return both **product**s and **reactant**s for such a reaction. An XPath address expressing this might be

```
//reaction/reactant |
                    //reaction[@type=′reversible′]/product
```

If the reversibility of the reaction is unknown, this attribute is not included.

Simple reaction topologies are defined with the **scheme** attribute on a **reactionStepList**. The steps in a **reactionStepList** with **scheme=“consecutive”** take place in sequence, i.e., one or more products of a reaction *n* may used in reaction *n*+1 or later but not earlier. For reactions taking place in parallel **scheme="simultaneous"**. Note that this attribute applies to the complete **reactionStepList**. If more complex topologies are required **reactionScheme** must be used.

3.3.3. Mechanism and mechanismComponent. The remaining elements describe properties of individual reactions such as reactive centers, mechanisms, and spectators. A ***mechanism*** is described by the IUPAC Gold Book[10] as *"a detailed description of the process leading from the reactants to the products of a reaction, including a characterization as complete as possible of the composition, structure, energy and other properties of reaction intermediates, products and transition states. An acceptable mechanism of a specified reaction (and there may be a number of such alternative mechanisms not excluded by the evidence) must be consistent with the reaction stoichiometry, the rate law and with all other available experimental data, such as the stereochemical course of the reaction. Inferences concerning the electronic motions which dynamically interconvert successive species along the reaction path (as represented by curved arrows, for example) are often included in the description of a mechanism."*

The Gold book also notes that "*for many reactions all this information is not available and the suggested mechanism is based on incomplete experimental data. It is not appropriate to use the term mechanism to describe a statement of the probable sequence in a set of stepwise reactions. That should be referred to as a reaction sequence, and not a mechanism.*"

With this in mind, CMLReact provides a **reactionScheme** and annotations to describe a reaction sequence, and both these and **mechanism** can co-occur within a **reactionScheme** container. These are unstructured elements to hold the mechanism of a reaction. This may be a simple textual description or a reference within a controlled vocabulary. Alternatively it may describe the complete progress of the reaction, including topological or Cartesian movement of atoms, bonds, and electrons as well as annotation with varying quantities, e.g. energies. For named reactions, e.g. Diels−Alder, Claisen rearrangement, aldol condensation, the name element should be used. When the "mechanism" is essentially a classification, e.g. "hydrolysis", the **label** element is more appropriate. In **reactionSchemes** there can be individual **mechanisms** for components of the reaction. In these cases (e.g. details of a specific proton transfer) the (new) element **mechanismComponent** should be used to refer to these components and also to add annotation. These **mechanismComponent**s can represent both physical constituents of the reaction or abstract concepts (types of bonds cleaved, thermodynamics, etc.). There are many ways in which these information components can be annotated within a reaction. One approach could be to refer to specific bonds and atoms through their **id**s and use **mechanismComponent** to describe their role, properties, etc. Another might be to use **mechanismComponent** to identify *types* of bond formed/broken without reference to actual atoms and bonds. Yet another could include information on the reaction profile.

3.3.4. spectator and spectatorList. This are objects in a reaction that are present during the course of a reaction but not formally involved in bond formation or cleavage. A spectator may be a catalyst but may also be an object that in some way constrains or facilitates the reaction. For example an amino acid residue may hydrogen bond to the substrate to make the reaction more favorable but is not itself changed during the course of that reaction. A molecule may be a spectator in one step but a reactant in another. The definition of a spectator is only relevant within a reaction. **spectator**s should occur within the **spectatorList** container.

3.3.5. transitionState. This contains details of a transition state such as 2-D and/or 3-D coordinates. It may contain partial bonds (which are not yet formalized in CML).

3.3.6. reactiveCenter. This element describes the set of bonds, atoms, and electrons involved in a reaction. While the semantics of the element are flexible, a recommended usage is to create **atomSet**(s) and **bondSet**(s) which map onto the groups (molecules) that undergo changes.

From domains such as reactions and computational chemistry, CML now has a requirement for atom and bond types. These describe classes of atoms and bonds which are not algorithmically deducible from their **elementType** or **order**. The **atomType** usually depends on the chemical or geometrical environment of the atom and is frequently assigned by algorithms with chemical perception, although some still have to be assigned by humans. **bondType**s are required to describe bonds in force fields, functional groups, reactions, etc.

3.4. Reaction Types. We have considered whether the language should contain explicit markup for reaction types (a referee who suggested that stereochemical changes such as "inversion of configuration" should be part of the language is thanked for this suggestion). Our approach is to instead allow the author to annotate the reaction with terms from formally agreed classifications such as the IUPAC gold book, and we have built a dictionary of some 130 such terms. This dictionary has been used to annotate 500 reactionSteps in the MACiE database.[11]

The basic unit of annotation is the <reaction>, but it is also possible to define the <reaction Center> and add annotation to it. If this is not sufficiently precise, then individual atoms can be annotated with RDF triples using the IUPAC or similar classification.

Some reaction types are very closely bound to the change in the hyperconnection table for the reaction. These include changes in

- oxidation state
- atom-based stereochemistry
- bond-based stereochemistry
- bond orders
- atom-based bond-order sum
- number and size of disjoint graphs
- number or type of ligands to an atom.

An example of how a reaction type can be deduced from the algorithmically determined change in the reaction is elaborated in more detail in example 10 below.

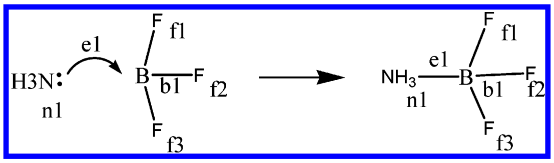


Figure 2. Reaction for example 1a.

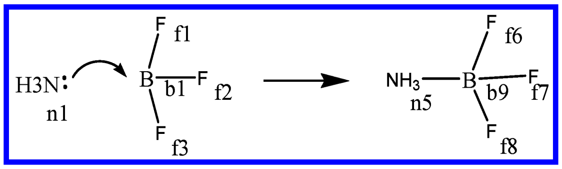


Figure 3. Reaction for example 1b.

4. EXAMPLES

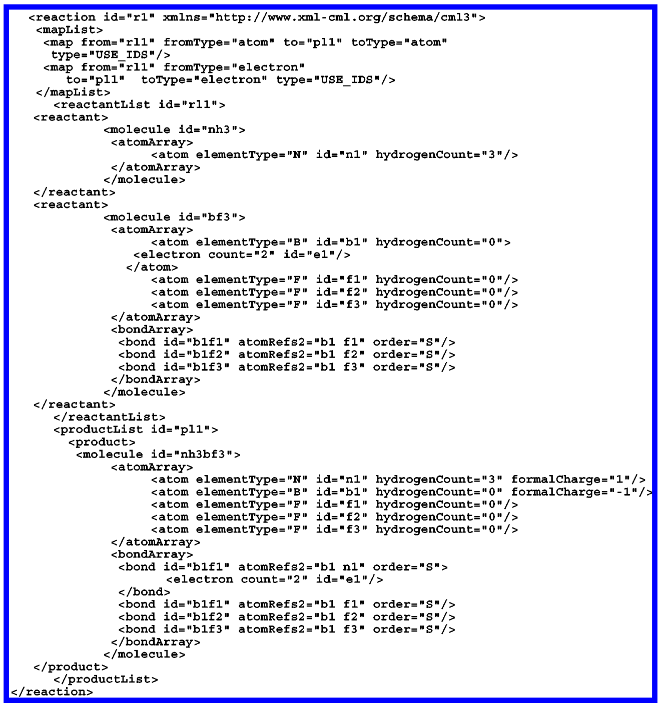
The examples show a wide but not complete range of the semantics used in CMLReact. In several examples, generic molecules (A, B, ...) are used and can obviously be replaced with either complete **molecule** elements or references to them.

4.1. Example 1. Complete Single Reaction with Atom Mapping. Two methods for atom mapping are given: (a) using identical atom/electron **id** values in **reactant** and **product** (Figure 2) and (b) using explicit mapping with **map** and **link** (Figure 3). A third approach using CMLSnap is given in example 9.

(a) **reactant** and **product** with identical locally unique atom **id**s. The mapping is indicated by the map elements indicating the local parent elements and the type of the mapped elements. The reserved attribute value **USE_IDS** indicates that the mapping can be constructed between species with identical ids implied by the XPath expressions

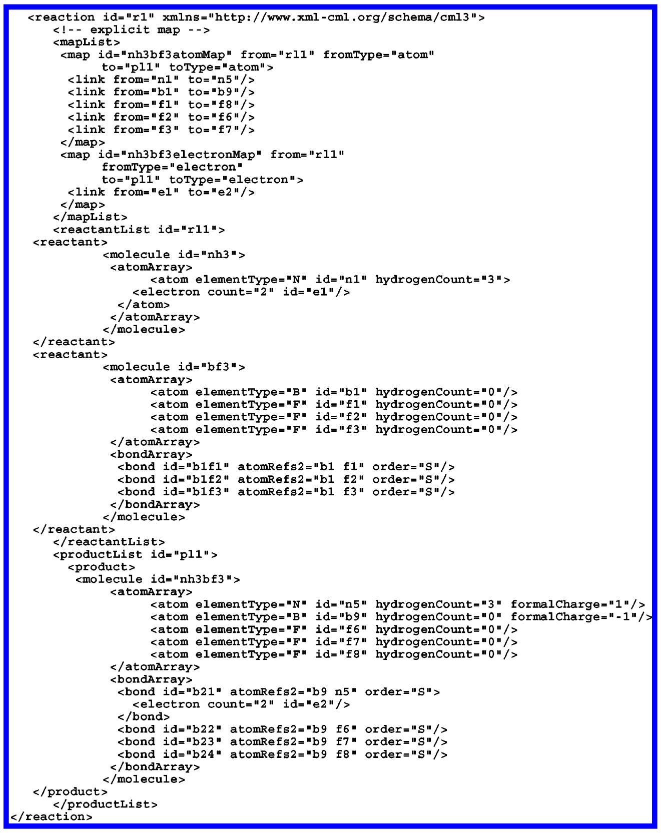
[@id=′rl1′]//atom/@id=[@id=′pl1′]//atom/@id

[@id=′rl1′]//electron/@id=[@id=′pl1′]//electron/@id



```
  <reaction id="r1" xmlns="http://www.xml-cml.org/schema/cml3">
   <mapList>
    <map from="rl1" fromType="atom" to="pl1" toType="atom"
     type="USE_IDS"/>
    <map from="rl1" fromType="electron"
       to="pl1"  toType="electron" type="USE_IDS"/>
   </mapList>
     <reactantList id="rl1">
  <reactant>
           <molecule id="nh3">
            <atomArray>
                 <atom elementType="N" id="n1" hydrogenCount="3"/>
            </atomArray>
           </molecule>
  </reactant>
  <reactant>
           <molecule id="bf3">
            <atomArray>
                 <atom elementType="B" id="b1" hydrogenCount="0">
               <electron count="2" id="e1"/>
              </atom>
                 <atom elementType="F" id="f1" hydrogenCount="0"/>
                 <atom elementType="F" id="f2" hydrogenCount="0"/>
                 <atom elementType="F" id="f3" hydrogenCount="0"/>
            </atomArray>
            <bondArray>
             <bond id="b1f1" atomRefs2="b1 f1" order="S"/>
             <bond id="b1f2" atomRefs2="b1 f2" order="S"/>
             <bond id="b1f3" atomRefs2="b1 f3" order="S"/>
            </bondArray>
           </molecule>
  </reactant>
     </reactantList>
     <productList id="pl1">
       <product>
        <molecule id="nh3bf3">
            <atomArray>
                 <atom elementType="N" id="n1" hydrogenCount="3" formalCharge="1"/>
                 <atom elementType="B" id="b1" hydrogenCount="0" formalCharge="-1"/>
                 <atom elementType="F" id="f1" hydrogenCount="0"/>
                 <atom elementType="F" id="f2" hydrogenCount="0"/>
                 <atom elementType="F" id="f3" hydrogenCount="0"/>
            </atomArray>
            <bondArray>
             <bond id="b1f1" atomRefs2="b1 n1" order="S">
                   <electron count="2" id="e1"/>
             </bond>
             <bond id="b1f1" atomRefs2="b1 f1" order="S"/>
             <bond id="b1f2" atomRefs2="b1 f2" order="S"/>
             <bond id="b1f3" atomRefs2="b1 f3" order="S"/>
            </bondArray>
           </molecule>
  </product>
     </productList>
</reaction>
```

(b) Using Explicit Maps. Within each **reactantList** and **productList** the atoms have unique ids, but these have no implicit relationship between them (Figure 3).



```
  <reaction id="r1" xmlns="http://www.xml-cml.org/schema/cml3">
     <!-- explicit map -->
     <mapList>
      <map id="nh3bf3atomMap" from="rl1" fromType="atom"
           to="pl1" toType="atom">
       <link from="n1" to="n5"/>
       <link from="b1" to="b9"/>
       <link from="f1" to="f8"/>
       <link from="f2" to="f6"/>
       <link from="f3" to="f7"/>
      </map>
      <map id="nh3bf3electronMap" from="rl1"
           fromType="electron"
           to="pl1" toType="electron">
       <link from="e1" to="e2"/>
      </map>
     </mapList>
     <reactantList id="rl1">
  <reactant>
           <molecule id="nh3">
            <atomArray>
                 <atom elementType="N" id="n1" hydrogenCount="3">
               <electron count="2" id="e1"/>
             </atom>
            </atomArray>
           </molecule>
  </reactant>
  <reactant>
           <molecule id="bf3">
            <atomArray>
                 <atom elementType="B" id="b1" hydrogenCount="0"/>
                 <atom elementType="F" id="f1" hydrogenCount="0"/>
                 <atom elementType="F" id="f2" hydrogenCount="0"/>
                 <atom elementType="F" id="f3" hydrogenCount="0"/>
            </atomArray>
            <bondArray>
             <bond id="b1f1" atomRefs2="b1 f1" order="S"/>
             <bond id="b1f2" atomRefs2="b1 f2" order="S"/>
             <bond id="b1f3" atomRefs2="b1 f3" order="S"/>
            </bondArray>
           </molecule>
  </reactant>
     </reactantList>
     <productList id="pl1">
       <product>
        <molecule id="nh3bf3">
            <atomArray>
                 <atom elementType="N" id="n5" hydrogenCount="3" formalCharge="1"/>
                 <atom elementType="B" id="b9" hydrogenCount="0" formalCharge="-1"/>
                 <atom elementType="F" id="f6" hydrogenCount="0"/>
                 <atom elementType="F" id="f7" hydrogenCount="0"/>
                 <atom elementType="F" id="f8" hydrogenCount="0"/>
            </atomArray>
            <bondArray>
             <bond id="b21" atomRefs2="b9 n5" order="S">
               <electron count="2" id="e2"/>
             </bond>
             <bond id="b22" atomRefs2="b9 f6" order="S"/>
             <bond id="b23" atomRefs2="b9 f7" order="S"/>
             <bond id="b24" atomRefs2="b9 f8" order="S"/>
            </bondArray>
           </molecule>
  </product>
     </productList>
</reaction>
```

4.2. Example 2. Complete Single Reaction with Annotations Showing Normalization through References (Figure 4).

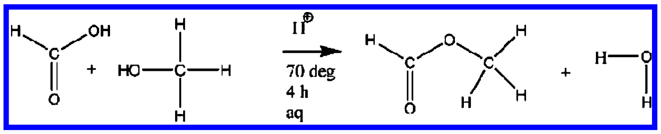
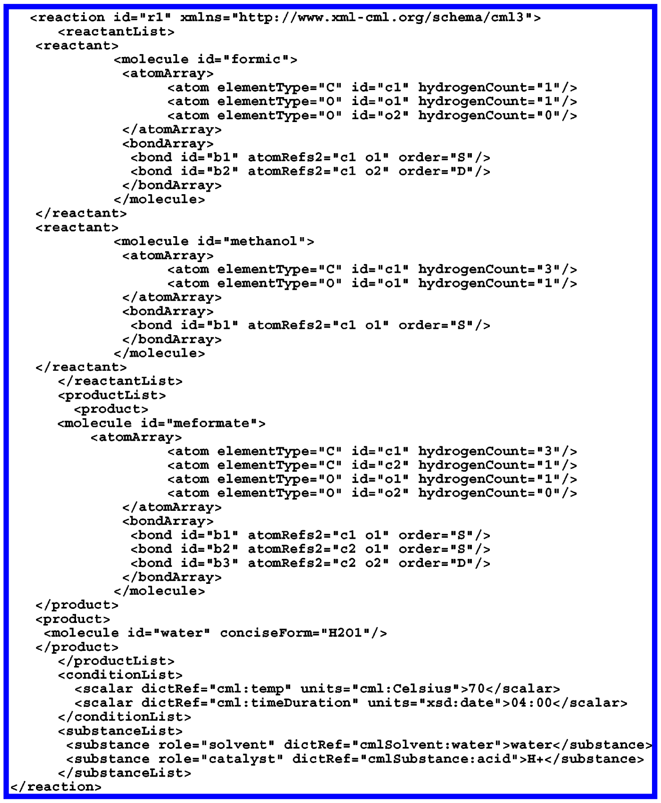
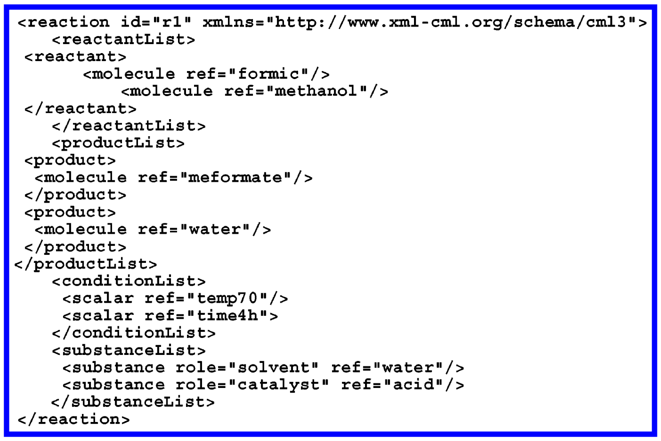


Figure 4. Reaction for example 2.



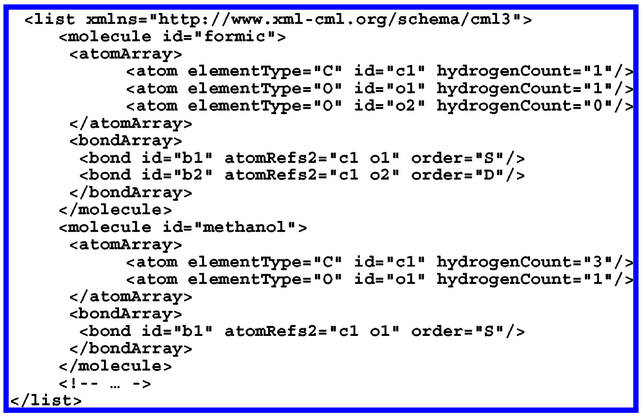
```
  <reaction id="r1" xmlns="http://www.xml-cml.org/schema/cml3">
     <reactantList>
  <reactant>
           <molecule id="formic">
            <atomArray>
                 <atom elementType="C" id="c1" hydrogenCount="1"/>
                 <atom elementType="O" id="o1" hydrogenCount="1"/>
                 <atom elementType="O" id="o2" hydrogenCount="0"/>
            </atomArray>
            <bondArray>
             <bond id="b1" atomRefs2="c1 o1" order="S"/>
             <bond id="b2" atomRefs2="c1 o2" order="D"/>
            </bondArray>
           </molecule>
  </reactant>
  <reactant>
           <molecule id="methanol">
            <atomArray>
                 <atom elementType="C" id="c1" hydrogenCount="3"/>
                 <atom elementType="O" id="o1" hydrogenCount="1"/>
            </atomArray>
            <bondArray>
             <bond id="b1" atomRefs2="c1 o1" order="S"/>
            </bondArray>
           </molecule>
  </reactant>
     </reactantList>
     <productList>
       <product>
     <molecule id="meformate">
          <atomArray>
                 <atom elementType="C" id="c1" hydrogenCount="3"/>
                 <atom elementType="C" id="c2" hydrogenCount="1"/>
                 <atom elementType="O" id="o1" hydrogenCount="1"/>
                 <atom elementType="O" id="o2" hydrogenCount="0"/>
            </atomArray>
            <bondArray>
             <bond id="b1" atomRefs2="c1 o1" order="S"/>
             <bond id="b2" atomRefs2="c2 o1" order="S"/>
             <bond id="b3" atomRefs2="c2 o2" order="D"/>
            </bondArray>
           </molecule>
  </product>
  <product>
   <molecule id="water" conciseForm="H2O1"/>
  </product>
     </productList>
     <conditionList>
       <scalar dictRef="cml:temp" units="cml:Celsius">70</scalar>
       <scalar dictRef="cml:timeDuration" units="xsd:date">04:00</scalar>
     </conditionList>
     <substanceList>
      <substance role="solvent" dictRef="cmlSolvent:water">water</substance>
      <substance role="catalyst" dictRef="cmlSubstance:acid">H+</substance>
     </substanceList>
</reaction>
```

This can be normalized to



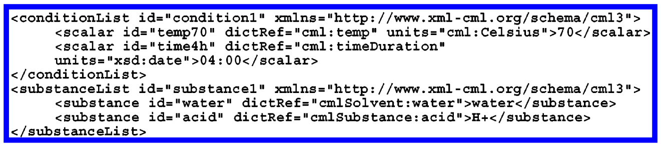
```
<reaction id="r1" xmlns="http://www.xml-cml.org/schema/cml3">
    <reactantList>
 <reactant>
       <molecule ref="formic"/>
            <molecule ref="methanol"/>
 </reactant>
    </reactantList>
    <productList>
 <product>
  <molecule ref="meformate"/>
 </product>
 <product>
  <molecule ref="water"/>
 </product>
</productList>
    <conditionList>
     <scalar ref="temp70"/>
     <scalar ref="time4h">
    </conditionList>
    <substanceList>
     <substance role="solvent" ref="water"/>
     <substance role="catalyst" ref="acid"/>
    </substanceList>
</reaction>
```

where the ref attributes point to a collection of molecules as in



```
  <list xmlns="http://www.xml-cml.org/schema/cml3">
      <molecule id="formic">
       <atomArray>
             <atom elementType="C" id="c1" hydrogenCount="1"/>
             <atom elementType="O" id="o1" hydrogenCount="1"/>
             <atom elementType="O" id="o2" hydrogenCount="0"/>
       </atomArray>
       <bondArray>
        <bond id="b1" atomRefs2="c1 o1" order="S"/>
        <bond id="b2" atomRefs2="c1 o2" order="D"/>
       </bondArray>
      </molecule>
      <molecule id="methanol">
       <atomArray>
             <atom elementType="C" id="c1" hydrogenCount="3"/>
             <atom elementType="O" id="o1" hydrogenCount="1"/>
       </atomArray>
       <bondArray>
        <bond id="b1" atomRefs2="c1 o1" order="S"/>
       </bondArray>
      </molecule>
      <!-- … ->
</list>
```

and conditions and solvents are treated in a similar manner. Note that the role attribute (e.g. on condition) remains on the main (referring) instance as its context cannot be abstracted into the referenced element.



```
<conditionList id="condition1" xmlns="http://www.xml-cml.org/schema/cml3">
     <scalar id="temp70" dictRef="cml:temp" units="cml:Celsius">70</scalar>
     <scalar id="time4h" dictRef="cml:timeDuration"
     units="xsd:date">04:00</scalar>
</conditionList>
<substanceList id="substance1" xmlns="http://www.xml-cml.org/schema/cml3">
     <substance id="water" dictRef="cmlSolvent:water">water</substance>
     <substance id="acid" dictRef="cmlSubstance:acid">H+</substance>
</substanceList>
```

4.3. Example 3. Schematic Reaction Expanded to CMLReact Components (Figure 5). The text above the arrow is a common convention for representing reactants belonging to the special class of "reagents" (Figure 5). The CMLReact representation allows not only for reagents to be identified but also for all reactants to be explicitly and equally treated.

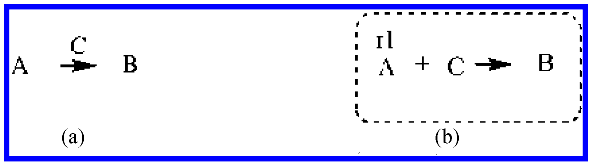
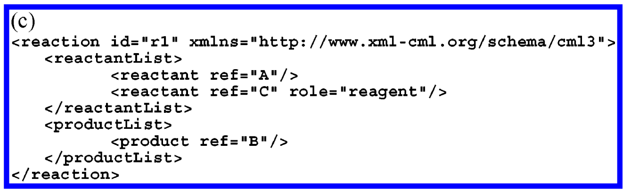


Figure 5. Schematic representation of example 3 where (a) is the traditional reaction representation and (b) is the representation as depicted by CMLReact.

In this and following examples the conventional representation is shown on the left (Figure 5a), the CML schematic CML architecture is shown on the right (Figure 5b), and the complete CML representation is shown below (c).



(c)

```
<reaction id="r1" xmlns="http://www.xml-cml.org/schema/cml3">
   <reactantList>
         <reactant ref="A"/>
         <reactant ref="C" role="reagent"/>
   </reactantList>
   <productList>
         <product ref="B"/>
   </productList>
</reaction>
```

4.4. Example 4. Consecutive Reactions Expanded to CMLReact Components (Figure 6). A **reactionStepList** contains consecutive steps where the product(s) of one are the reactant(s) of the next. The **type="consecutive"** attribute guarantees that the reactions are in a precise order.

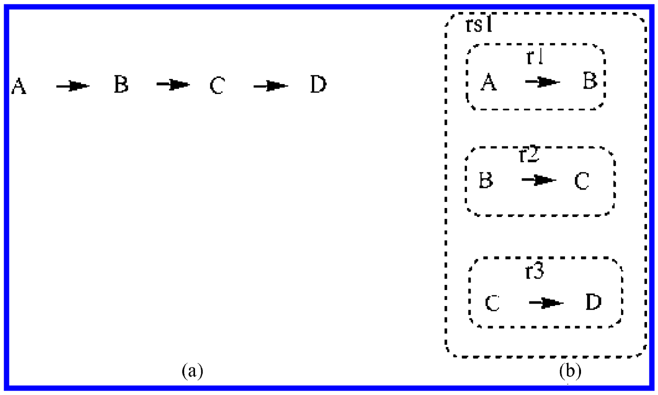
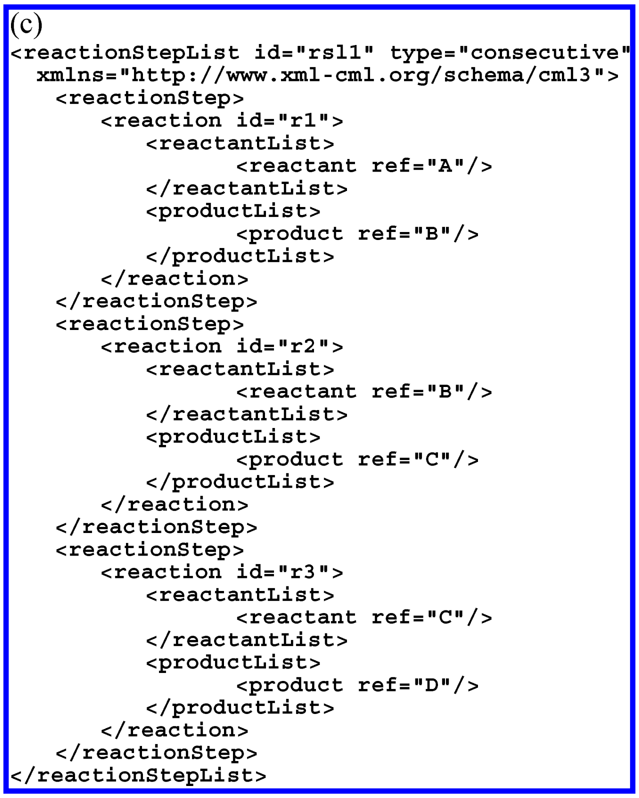


Figure 6. Schematic representation of example 4 where (a) is the traditional reaction representation and (b) is the representation as depicted by CMLReact.



(c)

```
<reactionStepList id="rsl1" type="consecutive"
  xmlns="http://www.xml-cml.org/schema/cml3">
   <reactionStep>
      <reaction id="r1">
         <reactantList>
                <reactant ref="A"/>
         </reactantList>
         <productList>
                <product ref="B"/>
         </productList>
      </reaction>
   </reactionStep>
   <reactionStep>
      <reaction id="r2">
         <reactantList>
                <reactant ref="B"/>
         </reactantList>
         <productList>
                <product ref="C"/>
         </productList>
      </reaction>
   </reactionStep>
   <reactionStep>
      <reaction id="r3">
         <reactantList>
                <reactant ref="C"/>
         </reactantList>
         <productList>
                <product ref="D"/>
         </productList>
      </reaction>
   </reactionStep>
</reactionStepList>
```

4.5. Example 5. A "Scheme" Representing Two Alternate Pathways for Chemical Processes (Figure 7). An overall **reactionScheme** consists of two subsidiary **reactionSchemes**. (An overall **reactionStepList** is not appropriate for branched topologies.) The individual schemes contain

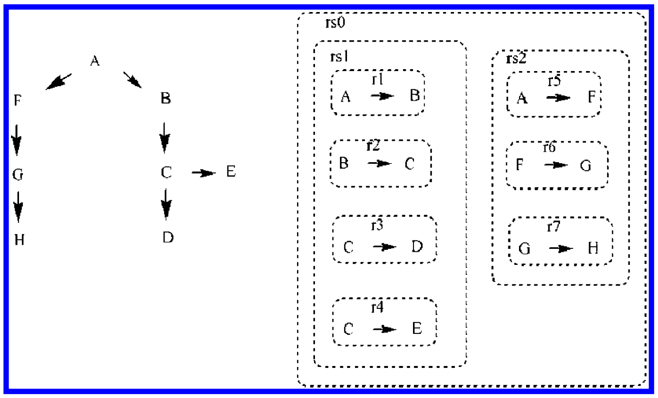
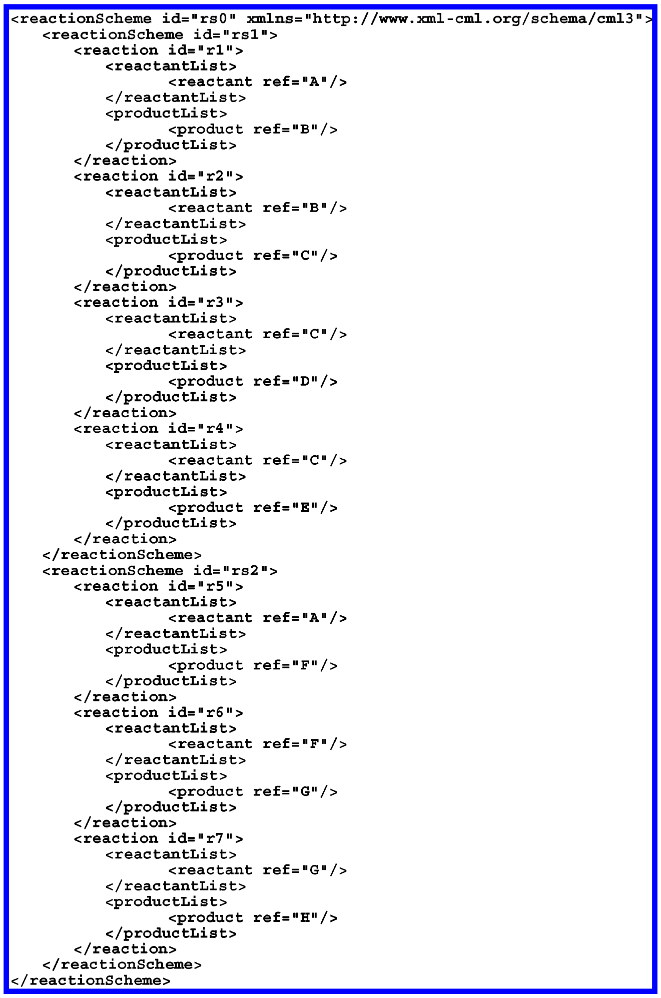


Figure 7. Schematic representation of example 5.

the normalized individual reactions, but the order is not completely defined, although it is helpful to list reactions later than other reactions on which they are dependent.



```
<reactionScheme id="rs0" xmlns="http://www.xml-cml.org/schema/cml3">
   <reactionScheme id="rs1">
      <reaction id="r1">
         <reactantList>
               <reactant ref="A"/>
         </reactantList>
         <productList>
               <product ref="B"/>
         </productList>
      </reaction>
      <reaction id="r2">
         <reactantList>
               <reactant ref="B"/>
         </reactantList>
         <productList>
               <product ref="C"/>
         </productList>
      </reaction>
      <reaction id="r3">
         <reactantList>
               <reactant ref="C"/>
         </reactantList>
         <productList>
               <product ref="D"/>
         </productList>
      </reaction>
      <reaction id="r4">
         <reactantList>
               <reactant ref="C"/>
         </reactantList>
         <productList>
               <product ref="E"/>
         </productList>
      </reaction>
   </reactionScheme>
   <reactionScheme id="rs2">
      <reaction id="r5">
         <reactantList>
               <reactant ref="A"/>
         </reactantList>
         <productList>
               <product ref="F"/>
         </productList>
      </reaction>
      <reaction id="r6">
         <reactantList>
               <reactant ref="F"/>
         </reactantList>
         <productList>
               <product ref="G"/>
         </productList>
      </reaction>
      <reaction id="r7">
         <reactantList>
               <reactant ref="G"/>
         </reactantList>
         <productList>
               <product ref="H"/>
         </productList>
      </reaction>
   </reactionScheme>
</reactionScheme>
```

4.6. Example 6. A Reaction with Implicit Steps and Anonymous Intermediates (Figure 8).

A is converted to B first by the addition of HCl and then in a later step by NaOH. This is often written as a list of annotations above the arrow (sometimes multiple arrows) with additional information in the caption. This is represented in two steps with an intermediate (X) where **role="anonymous"** indicates that it cannot be further expanded. Note that all products and reactants are uniquely identified, and both ref and id are used in this example. No dereferencing of anonymous elements is possible, and attempts to do so should throw exceptions.

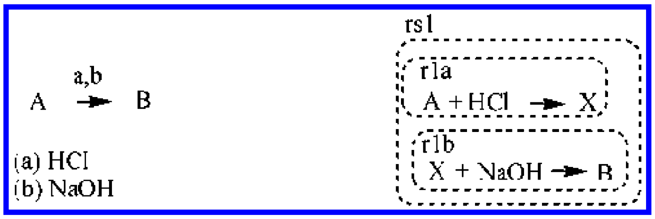
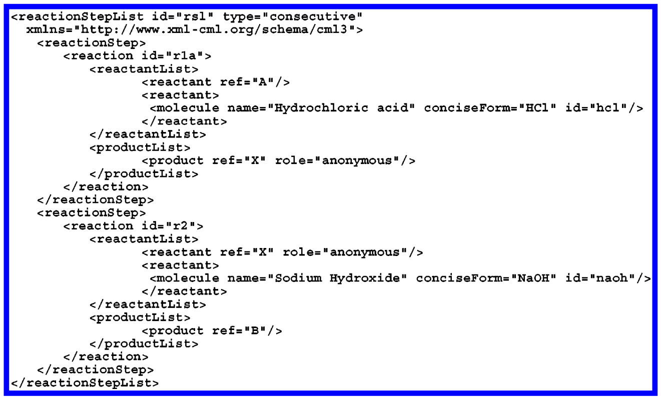


Figure 8. Schematic representation of example 6.



```
<reactionStepList id="rsl" type="consecutive"
  xmlns="http://www.xml-cml.org/schema/cml3">
   <reactionStep>
      <reaction id="r1a">
         <reactantList>
               <reactant ref="A"/>
               <reactant>
                <molecule name="Hydrochloric acid" conciseForm="HCl" id="hcl"/>
               </reactant>
         </reactantList>
         <productList>
               <product ref="X" role="anonymous"/>
         </productList>
      </reaction>
   </reactionStep>
   <reactionStep>
      <reaction id="r2">
         <reactantList>
               <reactant ref="X" role="anonymous"/>
               <reactant>
                <molecule name="Sodium Hydroxide" conciseForm="NaOH" id="naoh"/>
               </reactant>
         </reactantList>
         <productList>
               <product ref="B"/>
         </productList>
      </reaction>
   </reactionStep>
</reactionStepList>
```

4.7. Example 7. A "Branched" Reaction with Multiple Products (Figure 9).

A reacts to give both B and C (**type="simultaneous"**) whose ratio is given here. Note that the yield can also be given on either the **reactionStep**s or the overall **reactionStepList**. The semantics require that the reactants are identical in all **reactionStep**s.

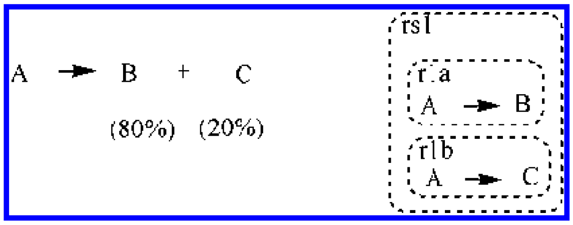
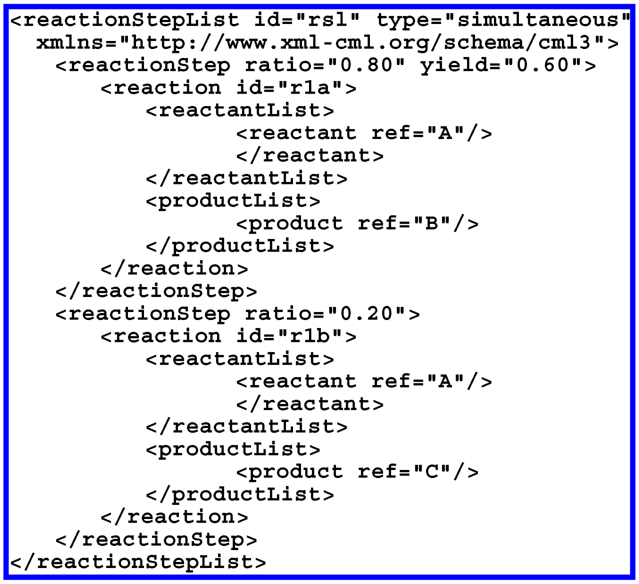


Figure 9. Schematic representation of example 7.



```
<reactionStepList id="rsl" type="simultaneous"
  xmlns="http://www.xml-cml.org/schema/cml3">
   <reactionStep ratio="0.80" yield="0.60">
      <reaction id="r1a">
         <reactantList>
               <reactant ref="A"/>
               </reactant>
         </reactantList>
         <productList>
               <product ref="B"/>
         </productList>
      </reaction>
   </reactionStep>
   <reactionStep ratio="0.20">
      <reaction id="r1b">
         <reactantList>
               <reactant ref="A"/>
               </reactant>
         </reactantList>
         <productList>
               <product ref="C"/>
         </productList>
      </reaction>
   </reactionStep>
</reactionStepList>
```

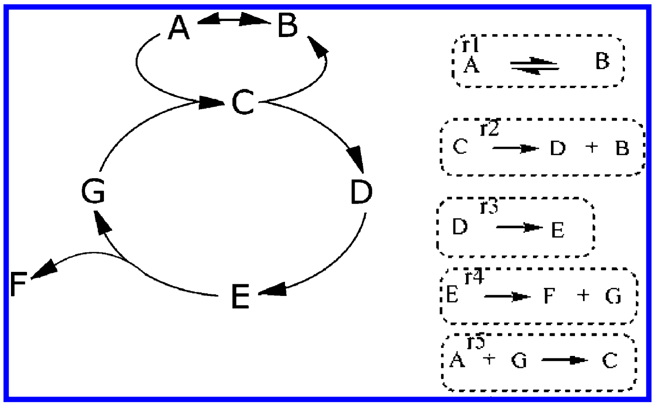
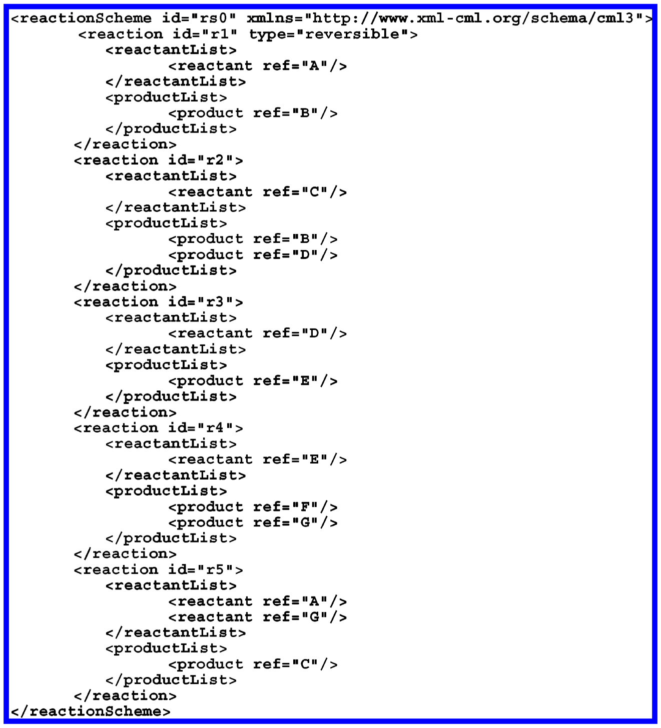


Figure 10. Schematic representation of example 8.

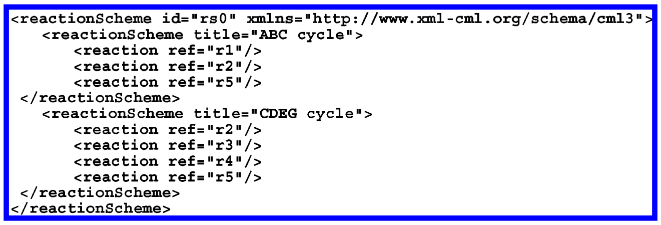
4.8. Example 8. A Complex Branched and Cyclic Reaction Scheme (Figure 10).

The reactions are in no definable order (though they may contain geometrical layout information). There is no explicit definition of the cycles though a semantic processor might be able to deduce this. The branched arrows do not here imply simultaneous reactions but rather that some of the products then react in different reactions. It is possible to split the system into two subcycles (ABC, CDEFG) which requires the reactions to be referenced rather than included explicitly in the reaction schemes.



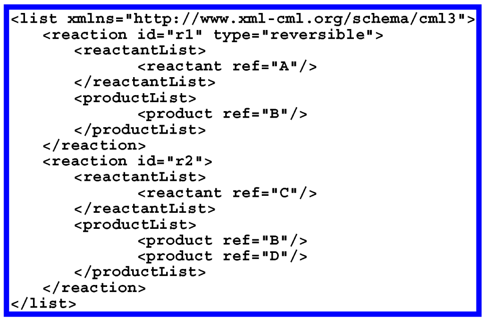
```
<reactionScheme id="rs0" xmlns="http://www.xml-cml.org/schema/cml3">
      <reaction id="r1" type="reversible">
         <reactantList>
               <reactant ref="A"/>
         </reactantList>
         <productList>
               <product ref="B"/>
         </productList>
      </reaction>
      <reaction id="r2">
         <reactantList>
               <reactant ref="C"/>
         </reactantList>
         <productList>
               <product ref="B"/>
               <product ref="D"/>
         </productList>
      </reaction>
      <reaction id="r3">
         <reactantList>
               <reactant ref="D"/>
         </reactantList>
         <productList>
               <product ref="E"/>
         </productList>
      </reaction>
      <reaction id="r4">
         <reactantList>
               <reactant ref="E"/>
         </reactantList>
         <productList>
               <product ref="F"/>
               <product ref="G"/>
         </productList>
      </reaction>
      <reaction id="r5">
         <reactantList>
               <reactant ref="A"/>
               <reactant ref="G"/>
         </reactantList>
         <productList>
               <product ref="C"/>
         </productList>
      </reaction>
</reactionScheme>
```

This could be normalized to



```
<reactionScheme id="rs0" xmlns="http://www.xml-cml.org/schema/cml3">
   <reactionScheme title="ABC cycle">
      <reaction ref="r1"/>
      <reaction ref="r2"/>
      <reaction ref="r5"/>
 </reactionScheme>
   <reactionScheme title="CDEG cycle">
      <reaction ref="r2"/>
      <reaction ref="r3"/>
      <reaction ref="r4"/>
      <reaction ref="r5"/>
 </reactionScheme>
</reactionScheme>
```

where the reactions are held in a list:



```
<list xmlns="http://www.xml-cml.org/schema/cml3">
   <reaction id="r1" type="reversible">
      <reactantList>
            <reactant ref="A"/>
      </reactantList>
      <productList>
            <product ref="B"/>
      </productList>
   </reaction>
   <reaction id="r2">
      <reactantList>
            <reactant ref="C"/>
      </reactantList>
      <productList>
            <product ref="B"/>
            <product ref="D"/>
      </productList>
   </reaction>
</list>
```

4.9. Example 9. An Ugi-like Approach, with All Atoms Represented in All Phases of a Multistep Reaction (Figure 11).

This was earlier published as “CMLSnap”,[11] where by saving the complete atomic components at each step (“snapshot”) the reaction can be easily animated. It can be easily used with chemical editors that have no explicit concept of reactants or products. For an *n*-step reaction, *n*+1 giant molecules are required with consistency of atom ids throughout and all atoms present in all steps. Apart from the first last species, all molecules function as both products and reactants. Note that snapshots with two or more molecules are held as a parent molecule with two or more children. A, B, and C are compound **molecule**s which each contain all the atoms in the system with consistent atom **id**s. The original CMLSnap publication had considerable implicit semantics; we have removed these so that standard CML software can process this representation.

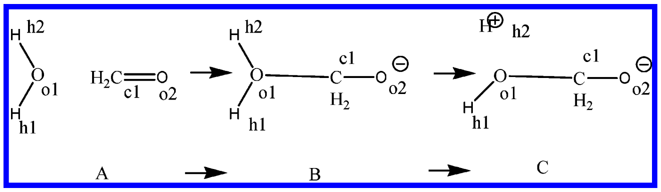
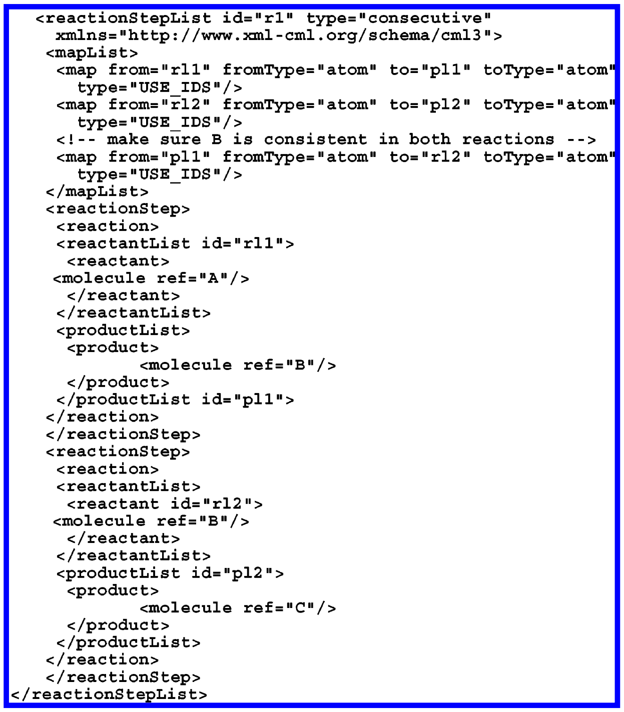


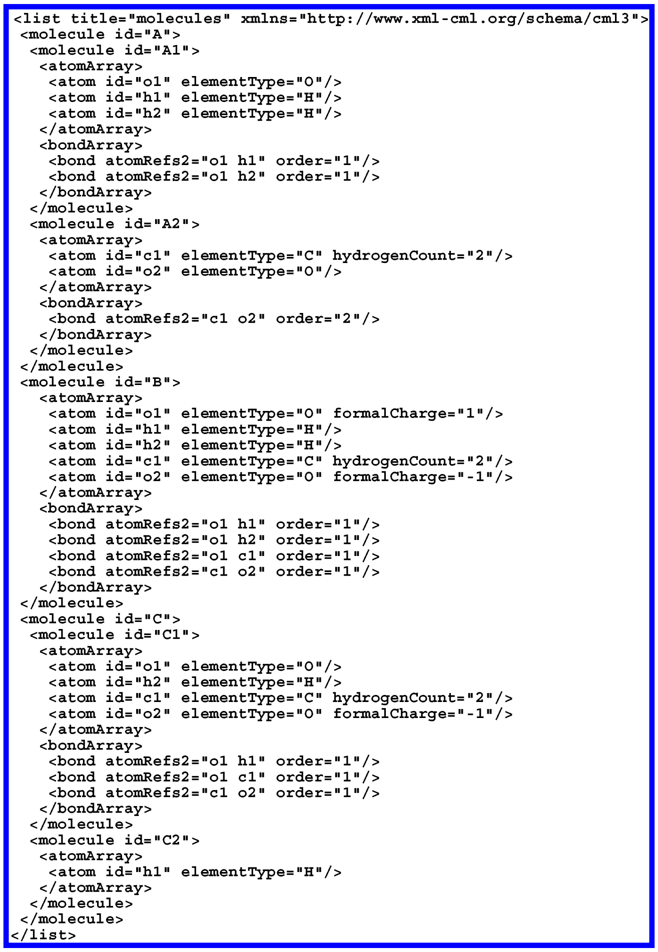
Figure 11. Schematic representation of example 9.

The full CML semantics would be



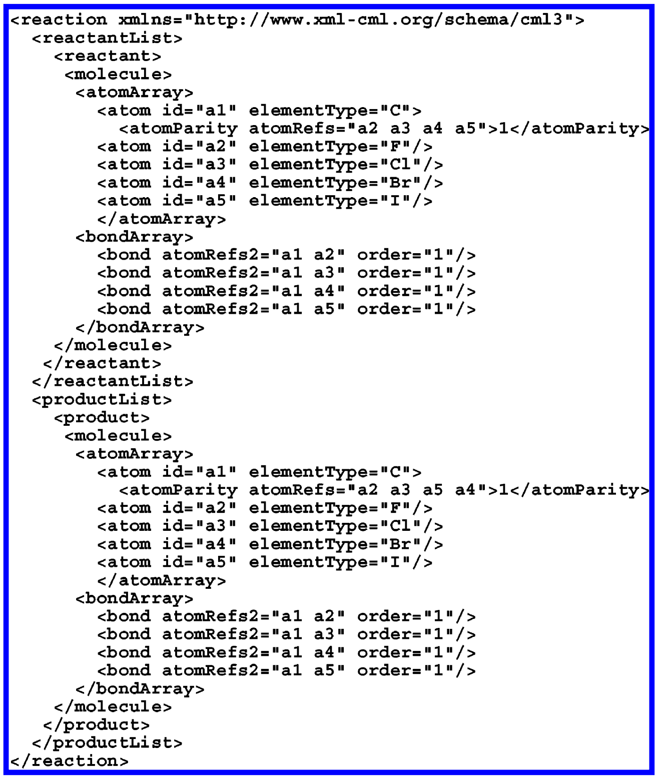
```
  <reactionStepList id="r1" type="consecutive"
    xmlns="http://www.xml-cml.org/schema/cml3">
   <mapList>
    <map from="rl1" fromType="atom" to="pl1" toType="atom"
      type="USE_IDS"/>
    <map from="rl2" fromType="atom" to="pl2" toType="atom"
      type="USE_IDS"/>
    <!-- make sure B is consistent in both reactions -->
    <map from="pl1" fromType="atom" to="rl2" toType="atom"
      type="USE_IDS"/>
   </mapList>
   <reactionStep>
    <reaction>
    <reactantList id="rl1">
     <reactant>
    <molecule ref="A"/>
     </reactant>
    </reactantList>
    <productList>
     <product>
            <molecule ref="B"/>
     </product>
    </productList id="pl1">
   </reaction>
   </reactionStep>
   <reactionStep>
    <reaction>
    <reactantList>
     <reactant id="rl2">
    <molecule ref="B"/>
     </reactant>
    </reactantList>
    <productList id="pl2">
     <product>
            <molecule ref="C"/>
     </product>
    </productList>
   </reaction>
   </reactionStep>
</reactionStepList>
```

where the reference targets are



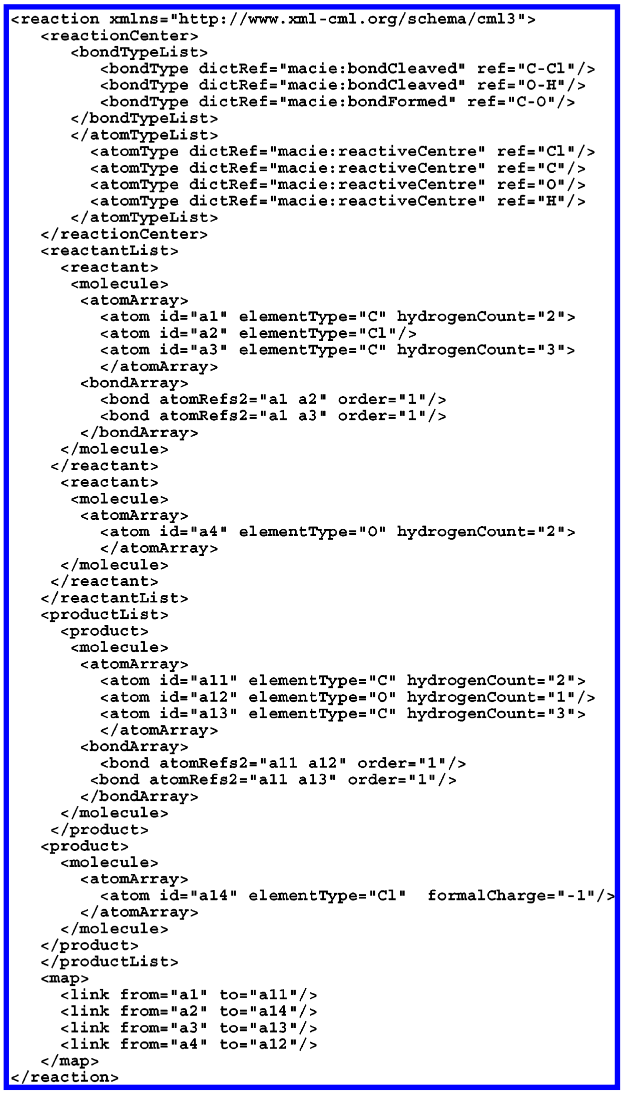
```
<list title="molecules" xmlns="http://www.xml-cml.org/schema/cml3">
 <molecule id="A">
  <molecule id="A1">
   <atomArray>
    <atom id="o1" elementType="O"/>
    <atom id="h1" elementType="H"/>
    <atom id="h2" elementType="H"/>
   </atomArray>
   <bondArray>
    <bond atomRefs2="o1 h1" order="1"/>
    <bond atomRefs2="o1 h2" order="1"/>
   </bondArray>
  </molecule>
  <molecule id="A2">
   <atomArray>
    <atom id="c1" elementType="C" hydrogenCount="2"/>
    <atom id="o2" elementType="O"/>
   </atomArray>
   <bondArray>
    <bond atomRefs2="c1 o2" order="2"/>
   </bondArray>
  </molecule>
 </molecule>
 <molecule id="B">
   <atomArray>
    <atom id="o1" elementType="O" formalCharge="1"/>
    <atom id="h1" elementType="H"/>
    <atom id="h2" elementType="H"/>
    <atom id="c1" elementType="C" hydrogenCount="2"/>
    <atom id="o2" elementType="O" formalCharge="-1"/>
   </atomArray>
   <bondArray>
    <bond atomRefs2="o1 h1" order="1"/>
    <bond atomRefs2="o1 h2" order="1"/>
    <bond atomRefs2="o1 c1" order="1"/>
    <bond atomRefs2="c1 o2" order="1"/>
   </bondArray>
 </molecule>
 <molecule id="C">
  <molecule id="C1">
   <atomArray>
    <atom id="o1" elementType="O"/>
    <atom id="h2" elementType="H"/>
    <atom id="c1" elementType="C" hydrogenCount="2"/>
    <atom id="o2" elementType="O" formalCharge="-1"/>
   </atomArray>
   <bondArray>
    <bond atomRefs2="o1 h1" order="1"/>
    <bond atomRefs2="o1 c1" order="1"/>
    <bond atomRefs2="c1 o2" order="1"/>
   </bondArray>
  </molecule>
  <molecule id="C2">
   <atomArray>
    <atom id="h1" elementType="H"/>
   </atomArray>
  </molecule>
 </molecule>
</list>
```

4.10. Reaction Typing Illustrated Using an Inversion of Configuration at an Atom. The following example represents inversion of configuration at atom a1 (an odd number of permutations of atom labels in the atomParity).



```
<reaction xmlns="http://www.xml-cml.org/schema/cml3">
  <reactantList>
    <reactant>
     <molecule>
      <atomArray>
        <atom id="a1" elementType="C">
          <atomParity atomRefs="a2 a3 a4 a5">1</atomParity>
        <atom id="a2" elementType="F"/>
        <atom id="a3" elementType="Cl"/>
        <atom id="a4" elementType="Br"/>
        <atom id="a5" elementType="I"/>
        </atomArray>
      <bondArray>
        <bond atomRefs2="a1 a2" order="1"/>
        <bond atomRefs2="a1 a3" order="1"/>
        <bond atomRefs2="a1 a4" order="1"/>
        <bond atomRefs2="a1 a5" order="1"/>
      </bondArray>
    </molecule>
   </reactant>
  </reactantList>
  <productList>
    <product>
     <molecule>
      <atomArray>
        <atom id="a1" elementType="C">
          <atomParity atomRefs="a2 a3 a5 a4">1</atomParity>
        <atom id="a2" elementType="F"/>
        <atom id="a3" elementType="Cl"/>
        <atom id="a4" elementType="Br"/>
        <atom id="a5" elementType="I"/>
        </atomArray>
      <bondArray>
        <bond atomRefs2="a1 a2" order="1"/>
        <bond atomRefs2="a1 a3" order="1"/>
        <bond atomRefs2="a1 a4" order="1"/>
        <bond atomRefs2="a1 a5" order="1"/>
      </bondArray>
    </molecule>
   </product>
  </productList>
</reaction>
```

The following elaboration represents a reaction center annotation with reference to the Macie dictionary.



```
<reaction xmlns="http://www.xml-cml.org/schema/cml3">
   <reactionCenter>
      <bondTypeList>
         <bondType dictRef="macie:bondCleaved" ref="C-Cl"/>
         <bondType dictRef="macie:bondCleaved" ref="O-H"/>
         <bondType dictRef="macie:bondFormed" ref="C-O"/>
      </bondTypeList>
      </atomTypeList>
        <atomType dictRef="macie:reactiveCentre" ref="Cl"/>
        <atomType dictRef="macie:reactiveCentre" ref="C"/>
        <atomType dictRef="macie:reactiveCentre" ref="O"/>
        <atomType dictRef="macie:reactiveCentre" ref="H"/>
      </atomTypeList>
   </reactionCenter>
   <reactantList>
     <reactant>
      <molecule>
       <atomArray>
         <atom id="a1" elementType="C" hydrogenCount="2">
         <atom id="a2" elementType="Cl"/>
         <atom id="a3" elementType="C" hydrogenCount="3">
         </atomArray>
       <bondArray>
         <bond atomRefs2="a1 a2" order="1"/>
         <bond atomRefs2="a1 a3" order="1"/>
       </bondArray>
     </molecule>
    </reactant>
     <reactant>
      <molecule>
       <atomArray>
         <atom id="a4" elementType="O" hydrogenCount="2">
         </atomArray>
     </molecule>
    </reactant>
   </reactantList>
   <productList>
     <product>
      <molecule>
       <atomArray>
         <atom id="a11" elementType="C" hydrogenCount="2">
         <atom id="a12" elementType="O" hydrogenCount="1"/>
         <atom id="a13" elementType="C" hydrogenCount="3">
         </atomArray>
       <bondArray>
         <bond atomRefs2="a11 a12" order="1"/>
        <bond atomRefs2="a11 a13" order="1"/>
       </bondArray>
     </molecule>
    </product>
   <product>
     <molecule>
       <atomArray>
         <atom id="a14" elementType="Cl"  formalCharge="-1"/>
       </atomArray>
     </molecule>
   </product>
   </productList>
   <map>
     <link from="a1" to="a11"/>
     <link from="a2" to="a14"/>
     <link from="a3" to="a13"/>
     <link from="a4" to="a12"/>
   </map>
</reaction>
```

From the map we know that

- a1 and a11 are “the same atom”
- a3 and a13 are “the same atom”
- a1 has lost ligand a2/a14 (Cl) (i.e. a1-a2 is broken)
- a1 has gained ligand a4/a12 (O) (i.e. a11-a12 is formed)

Software can then deduce that a C−Cl bond is replaced by a C−O bond and if required map this to another mechanistic representation such as an S_N2 reaction. The two examples could also be combined, including showing that the reaction is unbalanced (the loss of the proton).

5. CREATING AND PROCESSING CMLREACT

CML is designed to be a vendor-independent, nonexclusive specification, and an increasing number of chemical software and informatics providers are now using parts of it, though few yet support chemical reactions. We ask that implementations should be informed by the examples and should either process them in an acceptable manner or announce that some or all of the features of CMLReact have not been implemented. We would deprecate the creation of implementation-specific implied semantics.

Table 1. JUMBO Modules Supporting CMLReact and Its Semantics[a]

	ProductListTool	
FormulaTool	getAggregateFormula ()	gets aggregate formula for products
CMLAtom []	getProductAtoms ()	gets all descendant atoms
CMLBond []	getProductBonds ()	gets all descendant bonds
MoleculeTool []	getProductMolecules ()	gets all descendant molecules
	ProductTool	
FormulaTool	getFormula ()	gets formula for product
	ReactantListTool	
FormulaTool	getAggregateFormula ()	gets aggregate formula for reactants
CMLAtom []	getReactantAtoms ()	gets all descendant atoms
CMLBond []	getReactantBonds ()	gets all descendant bonds
MoleculeTool []	getReactantMolecules ()	gets all descendant molecules
	ReactantTool	
FormulaTool	getFormula ()	gets formula for reactant
	ReactionTool	
CMLList	mapReactantsToProducts (CMLReaction reaction, String control)	a list of possible mappings between all products and all reactants
void	mergeProductLists ()	merge productLists into single productList
void	mergeReactantLists ()	merge reactantLists into single reactantList
void	outputBalance (Writer w)	output and analyze aggregate formula for products and reactants
void	outputReaction (Writer w)	output simple inline version of reaction
void	partitionIntoMolecules ()	recursively splits reactant and product molecules
List	translateProductsToReactants ()	translate reactants and products geometrically to get the best fit
	SpectatorTool	
FormulaTool	getFormula ()	gets formula for spectator.
void	mergeChildMolecules()	merge child molecules A reaction may initially be created with the same spectator on both sides (such as pseudoreactant and pseudoproduct)

[a] The interface (e.g. **ProductListTool**) is followed by some signatures.

Our JUMBO toolkit is Open (available from http://sourceforge.net/projects/cml) and can serve as an application-independent CML engine, especially for reference. It aims to provide reference semantics for all CML elements.[2] We have collaborated closely with the OpenSource group developing the graphical chemical editor JChempaint[12] and the CDK toolkit[13] who work to be CML-conformant, and we are happy to work with other early adopters (publicly or privately).

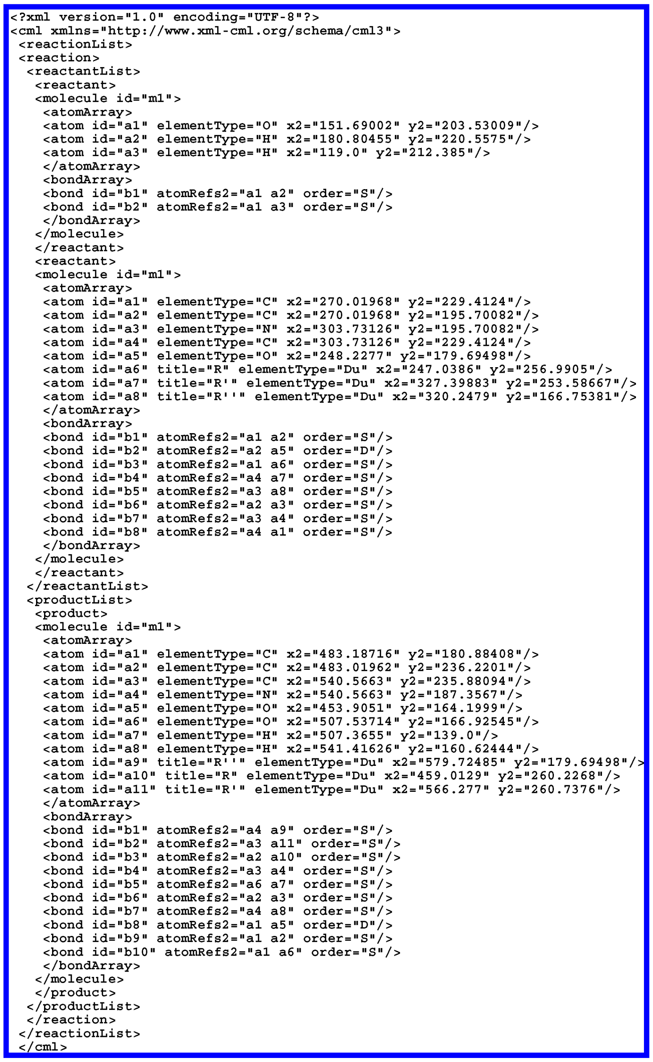
We note that publishers (though not yet the publisher of this journal) are increasingly keen to make the content of their full-text articles openly available. There is great scope for systematizing the capture of chemical reactions at time of authorship, including both the record or synthesis of individual molecules and more generally reactionSchemes of various granularity. The bioscience community is developing various levels of Systems Biology, and we have designed CMLReact with the expectation that it will be used to capture detailed systems biochemistry.[14]

5.1. XSL Stylesheets. In the modular approach we provide every element with a default stylesheet which renders the object to HTML and optionally SVG (Scalable Vector Graphics), and these are available for CMLReact, although the display is deliberately honest rather than aesthetic.

5.2. Tools. We have aimed to make sure that the language syntax is supportable and therefore briefly describe part of a typical program system. The tools in JUMBO directly relevant to CMLReact are as follows: **MapTool,** ProductTool, ProductListTool, ReactantTool, ReactantListTool, ReactionTool, ReactionListTool, and SpectatorTool. The functionality is, of course, non-normative for CML, but some examples of the nontrivial modules are shown in Table 1.

5.3. Editors and Legacy Conversion. A chemical editor may have an internal data structure which is not directly consistent with CMLReact concepts (e.g. the identification of product or reactant is based on geometry rather than explicit identification). We recommend the use of a CML-DOM for import and export[15] which requires the explicit internal conversion of these concepts. The DOM can then be serialized.

Using the manufacturers' published specifications, we have written Open legacy converters for files exported in the RXN and ChemDraw formats. These conversions, particularly from and to ChemDraw, are lossy as there are concepts which CML does not support or whose semantics are unclear. These converters use CMLDOM and are therefore useful models for implementations. The following shows an RXN file imported into JChemPaint and its export as CML:



```
<?xml version="1.0" encoding="UTF-8"?>
<cml xmlns="http://www.xml-cml.org/schema/cml3">
 <reactionList>
 <reaction>
  <reactantList>
   <reactant>
   <molecule id="m1">
    <atomArray>
    <atom id="a1" elementType="O" x2="151.69002" y2="203.53009"/>
    <atom id="a2" elementType="H" x2="180.80455" y2="220.5575"/>
    <atom id="a3" elementType="H" x2="119.0" y2="212.385"/>
    </atomArray>
    <bondArray>
    <bond id="b1" atomRefs2="a1 a2" order="S"/>
    <bond id="b2" atomRefs2="a1 a3" order="S"/>
    </bondArray>
   </molecule>
   </reactant>
   <reactant>
   <molecule id="m1">
    <atomArray>
    <atom id="a1" elementType="C" x2="270.01968" y2="229.4124"/>
    <atom id="a2" elementType="C" x2="270.01968" y2="195.70082"/>
    <atom id="a3" elementType="N" x2="303.73126" y2="195.70082"/>
    <atom id="a4" elementType="C" x2="303.73126" y2="229.4124"/>
    <atom id="a5" elementType="O" x2="248.2277" y2="179.69498"/>
    <atom id="a6" title="R" elementType="Du" x2="247.0386" y2="256.9905"/>
    <atom id="a7" title="R'" elementType="Du" x2="327.39883" y2="253.58667"/>
    <atom id="a8" title="R''" elementType="Du" x2="320.2479" y2="166.75381"/>
    </atomArray>
    <bondArray>
    <bond id="b1" atomRefs2="a1 a2" order="S"/>
    <bond id="b2" atomRefs2="a2 a5" order="D"/>
    <bond id="b3" atomRefs2="a1 a6" order="S"/>
    <bond id="b4" atomRefs2="a4 a7" order="S"/>
    <bond id="b5" atomRefs2="a3 a8" order="S"/>
    <bond id="b6" atomRefs2="a2 a3" order="S"/>
    <bond id="b7" atomRefs2="a3 a4" order="S"/>
    <bond id="b8" atomRefs2="a4 a1" order="S"/>
    </bondArray>
   </molecule>
   </reactant>
  </reactantList>
  <productList>
   <product>
   <molecule id="m1">
    <atomArray>
    <atom id="a1" elementType="C" x2="483.18716" y2="180.88408"/>
    <atom id="a2" elementType="C" x2="483.01962" y2="236.2201"/>
    <atom id="a3" elementType="C" x2="540.5663" y2="235.88094"/>
    <atom id="a4" elementType="N" x2="540.5663" y2="187.3567"/>
    <atom id="a5" elementType="O" x2="453.9051" y2="164.1999"/>
    <atom id="a6" elementType="O" x2="507.53714" y2="166.92545"/>
    <atom id="a7" elementType="H" x2="507.3655" y2="139.0"/>
    <atom id="a8" elementType="H" x2="541.41626" y2="160.62444"/>
    <atom id="a9" title="R''" elementType="Du" x2="579.72485" y2="179.69498"/>
    <atom id="a10" title="R" elementType="Du" x2="459.0129" y2="260.2268"/>
    <atom id="a11" title="R'" elementType="Du" x2="566.277" y2="260.7376"/>
    </atomArray>
    <bondArray>
    <bond id="b1" atomRefs2="a4 a9" order="S"/>
    <bond id="b2" atomRefs2="a3 a11" order="S"/>
    <bond id="b3" atomRefs2="a2 a10" order="S"/>
    <bond id="b4" atomRefs2="a3 a4" order="S"/>
    <bond id="b5" atomRefs2="a6 a7" order="S"/>
    <bond id="b6" atomRefs2="a2 a3" order="S"/>
    <bond id="b7" atomRefs2="a4 a8" order="S"/>
    <bond id="b8" atomRefs2="a1 a5" order="D"/>
    <bond id="b9" atomRefs2="a1 a2" order="S"/>
    <bond id="b10" atomRefs2="a1 a6" order="S"/>
    </bondArray>
   </molecule>
   </product>
  </productList>
  </reaction>
 </reactionList>
 </cml>
```

5.4. Analysis and Rendering. Because they have not been formalized, many of the potential uses of reactions have been neglected. Assuming that a significant amount of data can

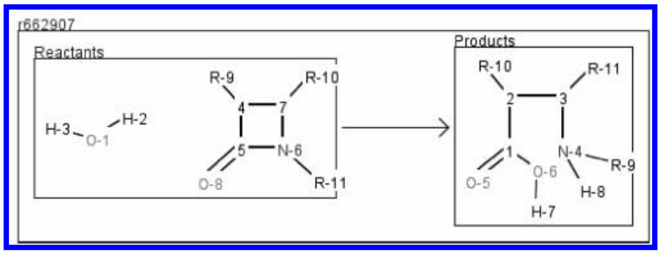


Figure 12. JChemPaint rendering of a typical reaction.[12]

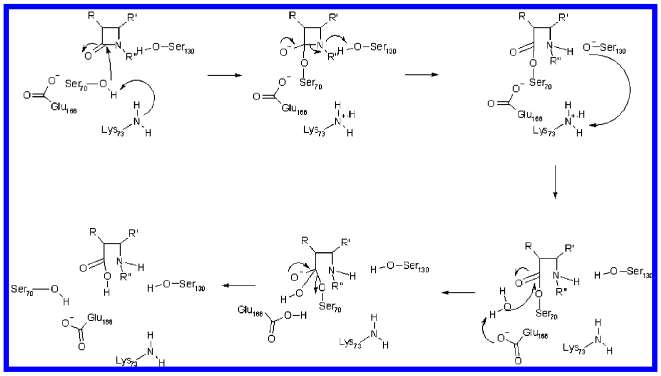


Figure 13. Traditional representation of a reaction scheme.

be made available in CMLReact it will make substantial improvements in the development of

- validating reaction data
- calculating formal electron movements ("curly arrows")
- integrating physical and observational data with reactions
- analysis of similarities and differences in reactions
- combining reactions
- analyzing multistep reactions and their topology
- capturing simulations of reactions and similar processes.[16]

Until now rendering has primarily been driven by 19th century paper-based concepts. We have explored the creation of dynamical digital objects to represent reactions and attach some automatically generated 2D "CMLSnaps" of multistep reactions. We urge that publishers start to embrace the potential of this new technology for the active capture of reaction content and its dissemination.

The traditional representation is shown in Figure 13.

An active version of this CMLSnap representation is also available.[16]

REFERENCES AND NOTES

(1) Murray-Rust, P.; Rzepa, H. S. Chemical Markup, XML, and the World Wide Web. 4. CML Schema. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 757−772. DOI: 10.1021/ci0256541.

(2) Wakelin, J.; Murray-Rust, P.; Tyrrell, S.; Zhang, Y.; Rzepa, H. S. CML Tools and Information Flow in Atomic Scale Simulations. *Mol. Simul.* **2005**, *31*, 315−322. DOI: 10.1080/08927020500065850.

(3) Details can be found at the following URL: http://cml.sourceforge.net/list/.

(4) Ingold, C. K. *Structure and Mechanism in Organic Chemistry*, 2nd ed.; Cornell University Press: Ithaca, NY, 1969; Chapters 5−15.

(5) As reviewed by Hendrickson, J. B.; Chen, L. *The Encyclopedia of Computational Chemistry;* Schleyer, P. v. R., Allinger, N. L., Clark, T., Gasteiger, J., Kollman, P. A., Schaefer, H. F., Scheriner, P. R., Eds.; J. Wiley & Sons: Chichester, U.K., 1998; Vol. 4, pp 2381−2402.

(6) Valdutz, G. *Modern Approaches to Chemical Reaction Searching;* Willett, P., Ed.; 1986; Gower: London, pp 202−220.

(7) (a) Chen, L. *Handbook of Chemoinformatics*; Gasteiger, J., Ed.; Wiley-VCH: 2003; pp 347−388. (b) Hendrickson, J. B. Descriptions of reactions: their logic and applications. *Recl. Trav. Chim. Pays-Bas* **1992**, *111*, 323−334. Hendrickson, J. B. Comprehensive System for Classification and Nomenclature of Organic Reactions. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 852−860. Hendrickson, J. B.; Miller, T. M. Reaction classification and retrieval. A linkage between synthesis generation and reaction databases. *J. Am. Chem. Soc.* **1991**, *113*, 902−910. Hendrickson, J. B.; Sander, T. L. The systematic definition of organic reactions. *Chem. Eur. J.* **1995**, *1*, 449−453. (c) Arens, J. F. A formalism for the classification and design of organic reactions. I. The class of (-+)n reactions. *Recl. Trav. Chim. Pays-Bas* **1979**, *98*, 155−161. A formalism for the classification and design of organic reactions. II. The classes of (+-)n+ and (-+)n- reactions. *Recl. Trav. Chim. Pays-Bas* **1979**, *98*, 395−399. A formalism for the classification and design of organic reactions. III. The class of (+-)nC reactions. *Recl. Trav. Chim. Pays-Bas* **1979**, *98*, 471−500. (d) Zefirov, N. S.; Baskin, I. I.; Palyulin, V. A. SYMBEQ Program and Its Application in Computer-Assisted Reaction Design. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 994−999. DOI: 10.1021/ci00020a038. (e) Ugi, I.; Bauer, J.; Bley, K.; Dengler, A.; Dietz, A.; Fontain, E.; Gruber, B.; Herges, R.; Knauer, M.; Reitsam, K.; Stain, N. Computer-assisted solution of chemical problems: a new discipline in chemistry. *Angew. Chem., Int. Ed.* **1993**, *32*, 201−227. Ugi, I.; Bauer, J.; Blomberger, C.; Brandt, J.; Dietz, A.; Fontain, E.; Gruber, B.; Scholley-Pfab, A. v.; Sneff, A.; Stein N. Models, concepts, theories, and formal languages in chemistry and their use as a basis for computer assistance in chemistry. *J. Chem. Inf. Comput. Sci.* **1994**, *34*, 3−16. DOI: 10.1021/ci00017a001.

(8) Murray-Rust, P.; Rzepa, H. S.; Tyrrell, S. M.; Zhang, Y. Representation and use of Chemistry in the Global Electronic Age. *Org. Biomol. Chem.* **2004**, *2*, 3192−3203. DOI: 10.1039/B410732.

(9) Stein, S. E.; Heller, S. R.; Tchekhovski, D. An Open Standard for Chemical Structure Representation − The IUPAC Chemical Identifier, Nimes International Chemical Information Conference Proceedings, 2003, pp 131−143.

(10) *IUPAC Compendium of Chemical Terminology ("Gold book")*; 2nd ed.; McNaught, A. D., Wilkinson, A., Eds; Blackwell Science: 1997.

(11) Holliday, G. L.; Mitchell, J. B. O.; Murray-Rust, P. CMLSnap. A novel method of reaction representation. *Internet J. Chem.* **2004**, *7*, Article 4.

(12) Krause, S.; Willighagen, E.; Steinbeck, C. JChemPaint − using the collaborative forces of the Internet to develop a free editor of 2D chemical structures. *Molecules* **2000**, *5*, 93−98. See, also: http://jchempaint.sourceforge.net/.

(13) Steinbeck, C.; Han, Y. Q.; Kuhn, S.; Horlacher, O.; Luttmann, E.; Willighagen, E. The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics. *J. Chem. Inf. Comput. Sci*. **2003**, *43*, 493−500. DOI: 10.1021/ci025584y See, also: http://cdk.sourceforge.net/

(14) Murray-Rust, P.; Mitchell, J. B. O.; Rzepa, H. S. Chemistry in Bioinformatics. *BMC Bioinformatics* **2005**, *6*, 141, DOI: 10.1186/1471-2105-6-141.

(15) P. Murray-Rust, P.; Rzepa, H. S. Chemical Markup, XML and the World-Wide Web. 2. Information Objects and the CMLDOM. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 1113−1123. DOI: 10.1021/ci000404a.

(16) Many examples of reaction animations can be found at http://www-mitchell.ch.cam.ac.uk/macie/.

CI0502698