Reaction SPL - extension of a public document markup standard to chemical reactions

Gunther Schadow[1], Yulia V. Borodina[2], Victorien Delannée[3], Wolf-Dietrich Ihlenfeldt[4], Alexander G. Godfrey[5], Marc C. Nicklaus[3]

[1]Pragmatic Data LLC; [2]US Food and Drug Admininstration; [3]National Cancer Institute, NIH; [4]Xemistry GmbH; [5]National Center for Advancing Translational Sciences, NIH

Abstract: There are numerous formats and data models for describing reaction-related data. However, each offers only a limited coverage of the multitude of information that can be of interest to a broad user base in the context of chemical reactions. Structured Product Labeling (SPL) is a robust yet fairly light public XML document standard. It uses a highly generic but usefully refinable data schema, which is, like a language, highly expressive. We are therefore presenting an extension of SPL to chemical reactions ("Reaction SPL"). This extension is designed to support chemical manufacturing processes, which include as a minimum the chemical reaction and the procedures and conditions to run it. We provide an overview of the SPL reaction specification structures followed by some examples of documents with reaction data: predicted single-step reactions, a two-step synthesis, an enzymatic reaction, an example how to represent a reaction center, a patent, and a fully annotated reaction with by-products. Special attention is given to a mechanism for atom-atom mapping of reactions as well as to the possibility to integrate Reaction SPL with laboratory automation equipment, in particular automated synthesis devices.

Keywords: chemoinformatics; reactions; document markup; semantic format; XML; Structured Product Labeling (SPL); reaction center; atom-atom mapping; automated synthesis

Introduction

The challenges involving standardized representations of molecules – both for small compounds and larger molecules – are well-known in the chemoinformatics community. Reaction data elevate these challenges to another level. Many reactions are being described comprising three molecules: two reactants and the product. However, other data included with reactions can have very different types. In fact, the different reaction data "stakeholders" such as chemoinformaticians, synthetic chemists, theoretical and computational chemists, ELN users and designers, developers of Computer-Aided Synthesis Design software, publishers, reaction database providers, patent lawyers, regulatory agencies such as FDA, etc. may each have a different idea what a "reaction" is.

Documents and data sets used in these specific contexts typically handle this wealth of information in a way, i.e. using a specific reaction data format, that is targeted at the local needs of the software or organization, thus is neither comprehensive for all possible needs nor optimally designed for general data exchange. We have therefore previously pointed out the need for comprehensive handling of reaction data. [1]

The idea of standardized data format is that *many* points of view and purposes are represented and can relate to each other, without an insistence on details some just do not care about, and without inhibiting others to express all the details they do care about. We should want a format that support the full life cycle without barriers from R&D experiment, publications, patents, documentation and control of the production process, regulatory applications, quality monitoring, to trade and logistics; all types of reactions including tautomeric interconversions, catabolic reactions and chemical degradation; reactions executed in the hood or synthetic machinery as well as in living systems including entire reaction pathways; descriptive and prescriptive reaction information; single-step and multi-step reactions; and the "mood" of the reaction, such as whether it was successfully executed, attempted but failed, or a computer-aided prediction.

We are therefore proposing an extension of the Structured Product Labeling (SPL) standard to chemical reactions ("Reaction SPL"). SPL is based on the Health Level 7 (HL7) Reference Information Model

(RIM) [2]. One may ask, why is an information model coming from the health field being used for reactions? The reason is that information in the health field encompasses a very broad range of types of documents, data types, and degrees of formalized vs. free-text descriptions. SPL as a fairly light and robust XML document standard uses a highly generic data schema, and has seen use cases such as people, organizations, products, and devices; science and measurements, including complex data, waveforms, and imaging; missing data and uncertainty; workflows, protocols, and processes; and scale from geography down to organization, building, devices, substances, and molecules and their parts. To a good extent, chemical substance-type data, such as substance indexing SPL files published in NLM DailyMed [3], are therefore a subset of the world that HL7 describes. This concept is extended here to reactions and reaction-related data since reactions in the real world often have annotations that go beyond chemistry-type data.

This paper takes a brief stock of the kinds of reaction information that various existing formats currently represent. Warr [4] has recently presented some of the background of the development of HL7 and SPL as well as comparison with other formats and schemas for reaction representation as part of a report on the recent NIH Workshop on Reaction Informatics [5]. This paper then gives an overview of the SPL reaction specification structures and how they are used to represent such types of information. We present some real-world examples of documents with reaction data such as a patent and a paper that about the mechanism of a biocatalyst. This SPL extension is designed to support chemical manufacturing processes, which include as a minimum the chemical reaction and the procedures and conditions to run it but can support the full life cycle ranging from initial design through synthesis, publications, patents, documentation and control of the production process, regulatory applications and monitoring, and trade and logistics.

Special attention is given to a mechanism for atom-mapping of reactions as well as to the possibility to integrate Reaction SPL with laboratory automation equipment, both analytical equipment and, increasingly, automated synthesis devices. The comprehensive design of the HL7 RIM structures used in SPL allows micro- and macroscopic process scales to be represented in a comparable structure, and the design of timed and conditioned action plans as a Turing complete "programming language" can be effortlessly applied to the specification of automation processes.

Definition and Characteristics of Reaction SPL

Basic Characteristics of Reaction SPL

SPL is based on the HL7 Reference Information Model (RIM) [2], which is like a toolkit of reusable data elements that can be used for creating data models in different domains. It uses universal data types including such high-level types as “physical quantity” (PQ) with intervals and even probability distributions, and “general timing specification” conceptualized as a set for points in time (QSET<TS>) and many others. Physical quantities use the Unified Codes for Units of Measure (UCUM) [6] for units of measure. SPL has been designed such that it does not require modification of the XML schema for every extension although it is less generic than the RIM in an attempt to trade-off domain specificity and recognizable element names with generality. For example, the “processStep” element in SPL, which we use to represent reactions is just a refinement of the RIM class called “Act”, and the “interactor” element which is used to link reactants and products to the processStep is called “participation” in the RIM. Every expression in the SPL schema is also in the RIM, i.e. the SPL schema is a constraint and refinement of the RIM. SPL uses domain-specific terminologies. SPL use cases are quite easily described in domain-specific implementation guides, mostly using example XML “snippets” along with validation procedures that are spelled out in plain English and encoded in Schematron (XPath) assertions to be automatically testable. Thus, new use cases can be supported quickly without breaking the conceptual backbone model.

The Reference Information Mode (RIM) is a simple but powerful data schema, consisting of 5 top-level classes: Entity, Role, Participation, Act, and ActRelationship. It is like a general grammar for a language, consisting of nouns (Entities) and verbs (Acts) and grammatical glue to connect them. Entities represent physical objects, people, places, and things. Acts represent events, activities, interactions between the Entities participating in the Acts, linked to the Acts by Participation classes with a participation type specifying how the Entity participates in the Act (e.g., as performer, or subject, or consumable or durable material, etc.) The participations connect an Act not directly to an Entity but through a Role that is

“played” by the Entity. For example: a Person (Entity) plays a Patient (Role) and subject (Participation) of a Surgery (Act) performed (Participation) by a Surgeon (Role) played by another Person (Entity).

Roles not only have a player Entity but also a “scoper”, which is what recognized the player in this Role. For example, the Person playing the Patient role does that in the scope of a hospital (Entity), likewise the hospital recognized the doctor (Entity) as a surgeon (Role). Thus, Roles establish durable relationships. In the world of things Role includes very basic ontological and mereological relationships, for example, a wheel (Entity) plays a part (Role) of a car (Entity), the whole car scopes the “part” role played by the wheel. Other fundamental types of Role are the generalization/specialization relationship, the group/member relationship, the container/content relationship, and the mixture/ingredient relationship.

As Roles denote relationships between things ActRelationships connect Acts, and fundamental ActRelationship types are component, specialization, and instantiation, but also cause (end effect), reason (motivation), and many more. Acts can be thought of in different stages of realization; on the one side being an event that is just happening, and on the other extreme is an Act considered only as a potential action. And between these extremes exist Acts that are intended, desired, requested, promised, and planned, or taken as a conditional. This dimension is known in human languages as “mood” of verbs. By representing the mood as a dimension, the HL7 RIM becomes extremely expressive as an information and knowledge representation schema without becoming more complex, because there are still just these 5 basic classes with a small set of generally useful attributes.

This model has already been applied to the domain of chemical substances, and in this work, it is extended to Reactions. According to ISO IDMP substances may be specified in two opposite ways. The preferred way to specify a substance is by giving its chemical structure; however, for many medicinal substances, the exact structure may not be fully known or rather, there is no single chemical structure defining the substance, but it is a complex mixture of myriads of different molecules (structurally diverse). As an example, take “orange juice” or “coffee”. These substances can only be defined by telling essentially how they are made from their source materials, while source materials are usually a product of some process, the ultimate source materials are biologic organisms, e.g., the orange plant, the coffee plant. The organism here may be defined by its genome or more generally by a literature reference to the authority who described it. An orange fruit or a coffee bean is a part of the organism, but it is not as simply a part as a wheel is part of a car. The fruit is harvested at the right time when it is ripe, separated from the plant and then processed by pealing, squeezing, winnowing, and roasting, etc. This understanding of structurally diverse substances as the product of a derivation process, has prepared the entire RIM based SPL substance model to represent any processes involving substances, which then includes chemical reactions.

Substance Indexing SPL as the Basis for Reaction SPL

Substance indexing SPL files [8] are capable of describing a wide variety of substances ranging from small molecules to botanical abstracts. This capability will equally be present in Reaction SPL files though atomic description of the reactant and product molecules, or at least the part of it relevant for the reaction in the case of bio-macromolecules, will typically be required for a meaningful description of the reaction. One central part of the modular data model is that of a "moiety." A moiety can be any part of a substance. It does not have to be a complete functional group. It does not have to be covalently connected to other moieties. There are two types of moieties: Additive moiety, which contributes to a whole complex substance; Site of interest, which delineates features or sites of interests, such as amino acid connection points. The moiety is a "simple chemical" is used to describe small molecules. The structure of this moiety is represented by MOLFILE and/or SMILES. Small proteins and nucleic acids (up to 999 atoms and 999 bonds) are also represented as simple chemicals. Importantly, InChI is required for unique identification of the structure. InChI's canonical atom numbering is used as one central feature in the SPL description of molecules. For example, the definition of substituents relies on InChI canonical atom numbering (Fig. 1)

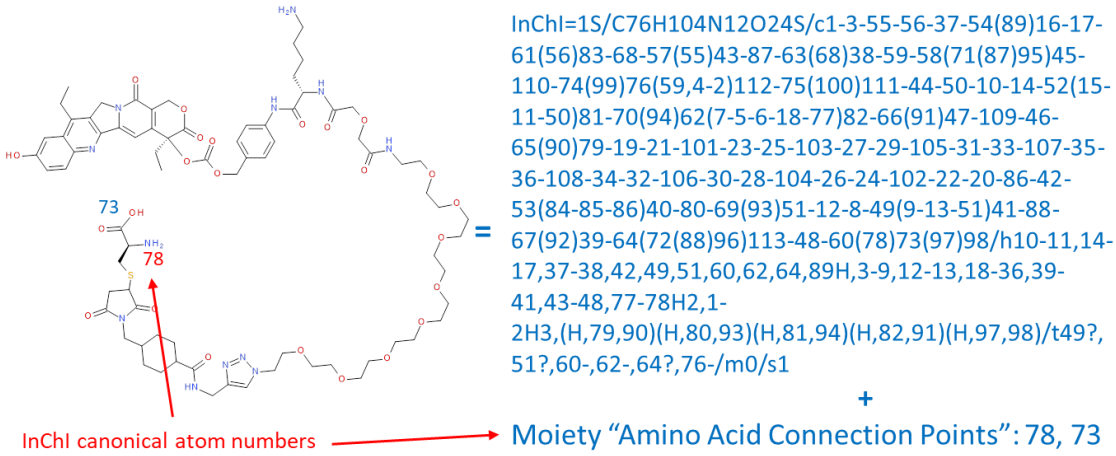


Figure 1. Definition of substituents via InChI canonical atom numbers.

Substance Indexing SPL files can convey information on a wide variety of substances, such as therapeutic proteins including modified proteins. Chemical modifications of proteins affect their biological activity and sometimes constitute the entirety of their biological activity. This includes modifications that occur as the result of natural biochemical processes, i.e posttranslational modifications. Substance Indexing SPL files describe such posttranslationally modified proteins with atomic precision even if one is dealing with a large protein [9]. We take advantage of these Substance Indexing SPL capabilities for Reaction SPL.

Reactions as Molecular Reaction Schemas

Reactants and reagents are specified with interactor participations elements of typeCode "consumable" (CSM). A functionCode can further say what some people call "role" in the reaction, such as "substrate", vs. "other reactant".

Products are specified with the interactor participations elements of typeCode "product" (PRD). A functionCode can label the main intended product vs. waste products (if they are even specified to balance the reaction.)

Any other agents in the reactions (that are typically written above the arrow, including catalysts and solvents, are specified with the interactor participations elements of typeCode "catalyst" (CAT), even if, in the case of a solvent, we would not consider that a "catalyst". Again, the functionCode can be used with domain specific terminology to say "catalyst" in the narrower sense vs. "solvent", or any other more specific designation. In practical terms, one can think of typeCode values as:

1. CSM - left side of arrow
2. PRD - right side of arrow
3. CAT -above and below the arrow
4. DIR - intermediate structures

("DIR" stems from "direct participant", i.e., a thing that is directly physically "somehow involved" but in this case not specified whether it is input or product or catalyst.) In this way, a reaction can be defined, including multi-step reactions by nested component/processSteps.

However, a full molecular specification of a reaction will often include more molecular details. One of them is the "atom-atom mapping" (AAM), which is to say which atoms from the reactants correspond with which atoms in the products. Then there is also the reaction center about which we may have information describing the detailed reaction mechanism. For example, a nucleophile attack by a negatively charged oxygen moiety on a partially positive charged C-atom whose electron is pulled by an electronegative halogen. These reaction mechanisms may be depicted with dashed and dotted pseudo bonds and arrows, in some cases conformation shifts may be depicted by multiple reaction center models as in a comic. In enzymatic reactions the structure of the reaction center and conformation shifts of the enzyme are often of great interest and might be detailed in 3D structures.

We have found that we can address molecular reaction center models using molfiles detailing how the reactants are aligned, if not sterically then at least schematically. We can use the molfile reaction center status value 4 to indicate bonds made or broken in the reaction. By connecting the reaction center model as to the reactants and products, we also achieve an atom-mapping requirement.

Comparison of Reaction SPL with Other Reaction Formats

It is worthwhile pointing out that Reaction SPL is really more a (formal) language than a chemical reaction format. Its highly generic data schema permits one to craft "sentences" (SPL documents) that span a virtually unlimited breadth of the types of info they can describe. Other formats may be better, and more-compact, for their specific tasks but Reaction SPL can cover pretty much everything that can be conceived. SPL, based on the very generic model HL7, must be made more specific for, e.g., reactions, which is (at least conceptually) straightforward. In contrast, InChI [7] for example is a very specific model, thus it is more difficult to broaden it to, e.g., general reaction descriptions. Nevertheless, it is useful to compare capabilities of specific formats for reaction handling and representation with the capabilities of Reaction SPL. We provide a tabular overview of this comparison in Table 1 (in the supplementary material ZIP file).

Types of Possible Reaction SPL Documents and Examples

Simple Reaction Schemas (Applied to Computer-Predicted Reactions)

The Synthetically Accessible Virtual Inventory (SAVI) project is an effort to generate a very large number of easily synthesizable molecules [10] that can be used in, e.g., drug design. About 1.75 billion molecules of the SAVI-2020 release with their associated proposed reactions can be downloaded from the freely accessible SAVI download page [11]. The Reaction SMILES string incorporated for every SAVI-2020 structure can be converted into RXN or RDF files. We have created a simple converter, a Unix shell script (included as rxr2rspl.sh in the supplementary material ZIP file), which converts RXN or RDF files to R-SPL files. This has been applied to a set of SAVI reactions. It is a simple transform of flat-structured RXN files into Reaction SPL (R-SPL) XML documents, with one reaction each. We present as an R-SPL file (as the file reaction-2875.xml in the supplementary material ZIP file) a single-step synthesis based on the copper[I]-catalyzed azide-alkyne cycloaddition ("click chemistry") reaction rule (having SAVI transform ID 2875). Several hundred more SAVI reactions are included in both RDF and R-SPL files in the subdirectory "rdf" in the supplementary material ZIP file, whose transforms can be looked up in https://cactus.nci.nih.gov/download/savi_download_transformwise/.

The interactor type codes are strictly only CSM (left side), PRD (right side of arrow), CSM (above arrow) and DIR (intermediates), but the function codes can be used to provide more specific codes for what the function of the reaction participant is, such as “substrate”, “reagent”, “product”, “by-product”, etc., defined by a terminology which can be easily extended:

```
<processStep>
  <interactor typeCode="CSM">
    <functionCode code="reactant" codeSystem="1.3.6.1.4.1.32366.1.1"/>
    <identifiedSubstance>
      ... substance definition elements molfile, InChI, etc. ...
    </identifiedSubstance>
  </interactor>
  <interactor typeCode="CSM">
    <functionCode code="reactant" codeSystem="1.3.6.1.4.1.32366.1.1"/>
    ...
  </interactor>
  <interactor typeCode="PRD">
    <functionCode code="product" codeSystem="1.3.6.1.4.1.32366.1.1"/>
    ...
  </interactor>
</processStep>
```

We note the atom numbering, which is included in the embedded CTAB blocks in the field mmm (14th column) in the Atom Block, in accordance with the CTfile format specification for reactions (such as RXN files) [12], which is how atom-atom mapping is achieved. While this feature is not strictly an R-SPL feature but simply gained by the fact that R-SPL provides for the encapsulation of molfiles, we do generally recommend that molfiles are always present, and InChIs, which can be generated from molfiles, and that the molfiles atom block is ordered according to the InChI atom numbering, which can be done from the AuxInfo output of InChI and the rxn2rspl.sh and mol2rspl.sh shell scripts perform this re-ordering. While SMILES or other representations are also supported, the reason we generally want molfiles and InChIs are that molfiles lend themselves to easy graphical representation (because they have atom coordinates), and InChIs atom numbers are used to reference to parts of the molecules with well defined numbers. And because molfiles have a special status for R-SPL, if molfiles provide the atom-mapping column, then the atom-mapping feature is served well by that mechanism and R-SPL does not need to replicate it.

There is an interesting aspect of predicted reaction libraries. The HL7 RIM on which SPL is based has a concept of (re-)act(ion) "mood", which is a code that indicates whether an act is actually occurring or planned or defined as a possible action. Among the SAVI reactions, only a very small number of them have so far been successfully carried out. While the "mood" could be used to indicate the hypothetical vs. actual feasibility status of the SAVI reaction, it is probably best to associate this attribute with a characteristic to the reaction. Characteristics are general parameter name – value pairs which can be defined to indicate different properties of the reaction, such as reversibility, enthalpy, etc. but also whether the reaction is predicted vs. has been observed (about actual reaction processes see further below.) In the current version of R-SPL, the mood code is not there to change in the actual SPL schema. We plan to add this in future releases.

Two-step Synthesis

The file reaction-isoxanzolines-2.xml in the supplementary material ZIP file shows a two-step reaction demonstrating the R-SPL capability of decomposing any action into sub-actions, in this case of "transforming" a reactant into the outcome of a preceding reaction.

```
<processStep>
  <component>
    <sequenceNumber value="1"/>
    <processStep>
      <interactor typeCode="CSM" ... reactant 1 .../>
      <interactor typeCode="CSM" ... reactant 2 .../>
      <interactor typeCode="PRD">
        <functionCode code="product" codeSystem="1.3.6.1.4.1.32366.1.1"/>
        <identifiedSubstance>
          <id extension="PZHSLBXRKIQQPV-ACCUITESSA-N" root="..."/>
          ... substance definition elements molfile, InChI, etc. ...
        </identifiedSubstance>
      </interactor>
      <interactor typeCode="PRD" ... product 2 .../>
    </processStep>
  </component>
  <component>
    <sequenceNumber value="2"/>
    <processStep>
      <interactor typeCode="CSM">
        <identifiedSubstance>
          <id extension="PZHSLBXRKIQQPV-ACCUITESSA-N" root="..."/>
        </identifiedSubstance>
      </interactor>
      <interactor typeCode="CSM" ... reactant 2 .../>
      <interactor typeCode="PRD" ... product 1 .../>
      <interactor typeCode="PRD" ... product 2 .../>
    </processStep>
  </component>
</processStep>
```

Since the product of the first reaction is the reactant of the second reaction, we do not need to repeat the definition of the substance when it repeats, instead, we assign an ID to it the first time and then we can reference it the second time through this ID.

Enzyme-driven Reaction

We are presenting a reaction driven by a biocatalyst to show how reaction SPL can handle catalysts and biomolecules, something that is clearly beyond the capability of molfiles and RXN. We are using one of the best studied enzymatic reactions, the lactate dehydrogenase (LDH) for which there are crystallographic structure models published by Cook and Senkovich [13] under the PDB accession numbers 4ND4 and 4ND3. The Reaction SPL file is "reaction-LDH-multistep.xml" in the supplementary material ZIP file. The LDH reaction is reversible,

pyruvate + NADH ↔ L-lactate + NAD^+ with the lactate dehydrogenase as the catalyst.

and in the direction of L-lactate as a product shows how stereo-selective enzymatic reactions happen by observing the details of the intermediate structures from the PDB files 4ND4 for the LDH-pyruvate-NADH complex and 4ND3 for the LDH-L-lactate-NAD^+ complex. The molfiles of the NAD ligand complexes were extracted with JSmol [14] from the original PDB entries, to reproduce the figure from ref. [13]. The NAD in the molfile of 4ND3 (pyruvate) was changed to NADH by manually editing 3 bond valences and having JSmol calculate hydrogens at the apex of NADH (atom position 2 in Figure 2). Then for each molfile the InChI identifier was generated and atoms in the molfile rearranged to follow the canonical InChI atom numbers.

These molfiles show how the pyruvate is strapped into place in one orientation by R171, N140, R199 and H195, and how the NAD is positioned by M163, I32, S99 and D53. This arrangement is what causes the LDH to synthesize L-lactate and not D-lactate.

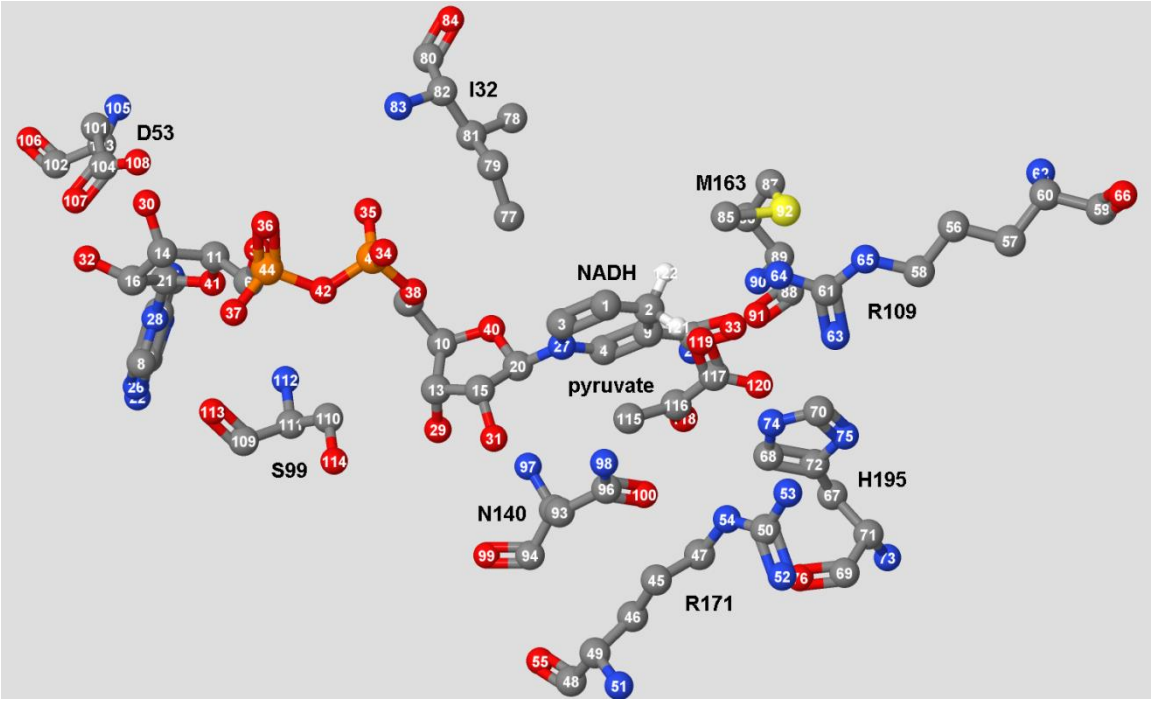


Figure 2: intermediate complex of LDH, NADH and pyruvate

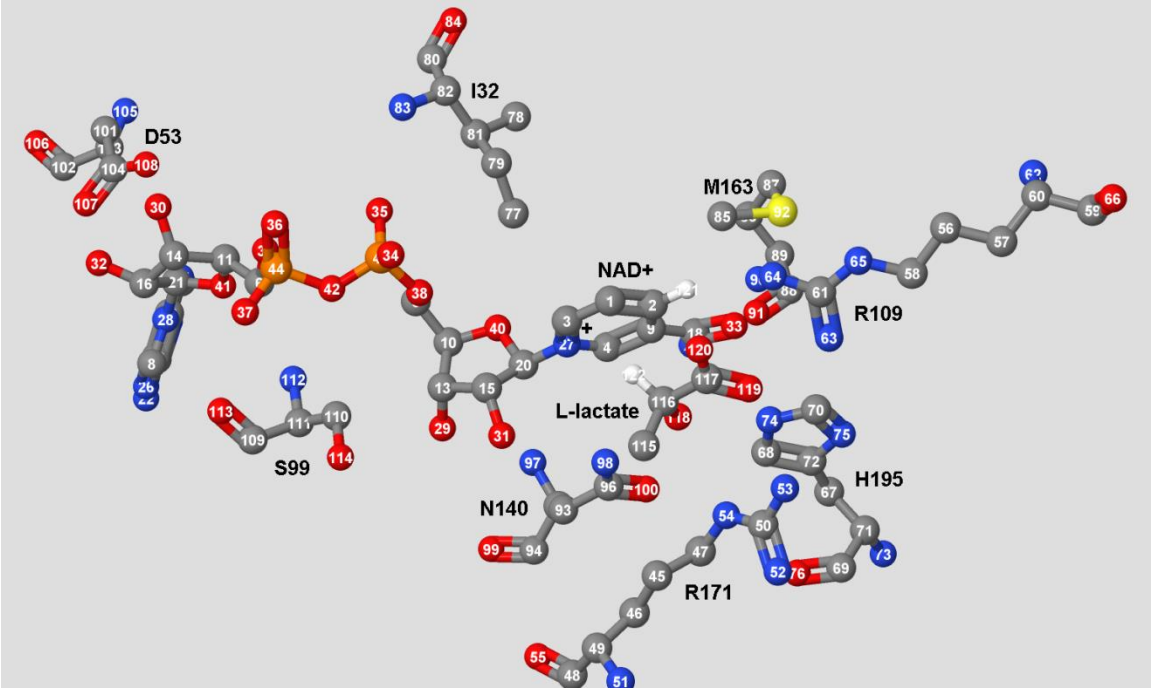


Figure 3: intermediate complex of LDH, NAD^+ and L-lactate

In the Reaction SPL file, all the reaction participants (interactors) are defined at the top level overall LDH reaction, where pyruvate and NADH (Fig. 2) is on the left side of the arrow (CSM), L-lactate and NAD^+ (Fig. 3) on the right side (PRD) and the LDH as the catalyst (CAT). In Reaction SPL we can then drill down into the reaction and establish at least 3 sub-steps:

1. LDH + pyruvate + NADH ---> LDH-pyruvate-NADH-complex
2. reacting of NADH + PYR in the complex to NAD^+ L-lactate
3. LDH-L-lactate-NAD^+-complex ---> LDH + L-lactate + NAD^+

Note that while the LDH is a catalyst of the overall reaction, once we drill down into sub-reaction steps, the catalyst becomes a reactant and finally a product. In most enzyme mechanisms the binding of ligands happens stepwise with specific conformation changes, all of which, if known or predicted, can be represented by these 3D molfiles of the critical components in additional intermediary steps. Of course, 2D molfiles can also be used if the exact coordinates are not known and the Reaction SPL author wants to visualize only schematic reaction mechanisms.

In order to keep the Reaction SPL file concise, once a substance is defined with molfile and InChI and other characteristics, it can be given an id, and references to it elsewhere are then very short only mentioning that id. It is useful to use customary labels such as “LDH”, or “PYR”, or “LAC” for these ids in order to keep the XML document readable.

Previously we had developed a technique to specify post-translational modifications to peptide sequences as substitutions of certain amino-acids by irregular substituents, with amino-acid connection points [9]. For reaction SPL we also developed an easier way to specify directly in the irregular substituent how it is inserted into the regular peptide chains. This leads to a more concise way of showing multiple intermediate structures without having to re-define the entire enzyme-ligand complex over and over again with all the chains and substitutions. But the principle is the same: a molfile is created with InChI and the atoms in the molfile sorted in InChI order to match the atom numbers (with all hydrogens – if any – at the end). This yields unambiguous canonical atom numbers used to define the amino-acid connection points, in pairs of amino-N and carboxy-C atom numbers. For example, I32 in the molfile has the amino-acid connection point at position (83, 80); and R109 at (62, 59) and so on. These connection points are then directly linked with a “bond” of type “amino-acid substitution site” to the chain by referencing the chain by its id (e.g.

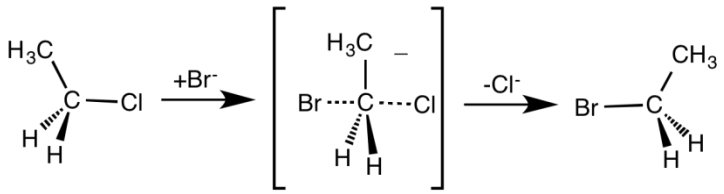
"LDH-A") and providing the position number in the sequence. And because this substitution method normally points from the biomolecules to the irregular structure substituted into it, the connection points pair is also referenced in the first position number. Since for brevity here we are allowing to connect the connection points directly into the peptide chain, the first position number is just the ordinal number of the current connection points pair, 1, 2, 3, and so on. So, for example, R109 is described as connecting to the LHA-A chain at position (1, 106), 1 because R109 is the first connection points moiety of our molfile (and it is the first one because its InChI atom number of the amino-N is the lowest of them all) and "106" is the position in the actual amino acid letter sequence which corresponds to what is labeled "R109" in the PDB file:

```
<identifiedSubstance>
  ... the 3d structure molfile, InChI, etc. ...
  <moiety>
    <code code="C118427" displayName="Amino-Acid Connection Points"
          codeSystem="2.16.840.1.113883.3.26.1.1"/>
    <positionNumber value="51"/>
    <positionNumber value="48"/>
    <partMoiety>
      <name>R109</name>
      <bond>
        <code code="C118426" displayName="Amino Acid Substitution Site"
              codeSystem="2.16.840.1.113883.3.26.1.1"/>
        <positionNumber value="1"/>
        <positionNumber value="106"/>
        <distalMoiety>
          <id extension="LDH-A" root="6ef20731-e18a-4f16-b952-6e0bc95f5931"/>
        </distalMoiety>
      </bond>
    </partMoiety>
  </moiety>
  ... additional connection points ...
</identifiedSubstance>
```

The discrepancy between "R109" being at the actual position 106 seems confusing, and we are left to guess why exactly there is such discrepancy. This may be because Cook et al. were trying to reference well known position that are conventionally labeled R109 even though their own sequence has this at position 106. But this indirection is a good test case to show how Reaction SPL gives the flexibility to use author-controlled labels (in the connection points moiety name "R109") and yet unambiguous correct index numbers into the actual sequences, which happens to be 106.

Reaction Center Expressed in R-SPL

Intermediate structures are often used to describe reaction mechanisms such as in textbook presentations of a nucleophile substitution:



Similar to these intermediate structures of a reaction mechanism there are Imaginary Transition Structures (ITS) [15] or Condensed Graphs of Reaction (CGR) [16], [17] that conceptualize a chemical reaction as one single pseudo molecule with incoming bonds and outgoing bonds. Delannée and Nicklaus have used this idea to create a novel reaction representation "ReactionCode" [18], which is a hierarchical code beginning from the reaction center and then continuing out from there. These presentations of reactions are alternatives to the arrow notation and have lots of benefits [19]. The arrow notation will probably never disappear in favor of reaction graph models, but since they are interconvertible and are very similar to reaction mechanism descriptions, Reaction SPL provides a framework for these different conceptualizations to coexist. This point has been made already in our enzymatic reaction description. But

now we generalize this for use with any molecular structures, no matter how large or small, polymeric, template-driven or random.

We show in the file reaction-center.xml (supplementary material ZIP file) the reaction center of a well-known substitution reaction (Fig. 4) [19], [20].

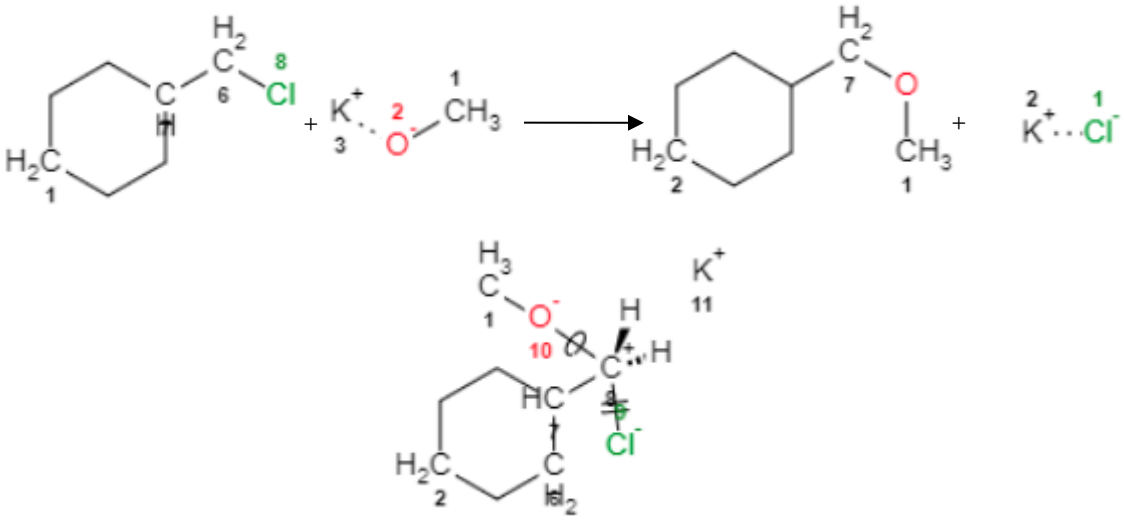


Figure 4: a substitution reaction with an imaginary transition structure.

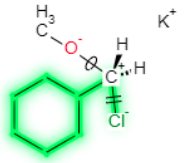
In this notation we show bonds that are made (in-bonds) as lines with an oval in the center, and bonds that are broken (out-bonds) as lines with a double line as in a strike-through, the notation which was first used by Fujita [15]. We also show partial charges as full charges. In a more mechanistic representation, the made bond might be shown as a dotted arrow indicating the nucleophile attack and the broken bond as a wedge indicating the electronegativity of the chlorine atom. Different ways of using, or extending, features of molfiles for indicating reaction center features are possible. Two of the authors (GS and WI) maintain their own molfile drawing code and adding these features would not be overly hard. Of course, this entails the risk that such molfiles cannot be parsed successfully by other software.

As in the enzyme example, the intermediate structures can be connected to the reactants with Chemical Structure Connection Points and Chemical Structure Substitution bonds. As in the amino acid connection points and substitution, we use InChI atom numbers to reference to these connection points. In case of a linear chain, it is simple to connect to the left and right side of each chain link. For random polymers this had already been extended to cover ladder polymers and branched polymers. Now we have extended the specification to allow for general structure substitution by specifying insertion points:

```
<identifiedSubstance>
  ... the intermediate structure molfile, InChI, etc. ...
  <moiety>
    <code code="Cxxxxxx" displayName="Chemical Structure Connection Points"
          codeSystem="2.16.840.1.113883.3.26.1.1"/>
    <positionNumber value="2"/>
    <positionNumber value="7"/>
    <positionNumber value="8"/>
    <positionNumber value="9"/>
    <partMoiety>
      <bond>
        <code code="Cyyyyyy" displayName="Chemical Structure Substitution"
              codeSystem="2.16.840.1.113883.3.26.1.1"/>
        <positionNumber value="1"/>
        <positionNumber value="1"/>
        <positionNumber value="7"/>
        <positionNumber value="6"/>
        <positionNumber value="8"/>
```

```
        <distalMoiety>
          <id extension="cmchx" root="c68fd29a-c9cd-4a47-8785-f1f9489f8cf9"/>
        </distalMoiety>
      </bond>
    </partMoiety>
  </moiety>
  ... other connection points with the other reaction participants ...
</identifiedSubstance>
```

The meaning of these positionNumbers is as follows: The Chemical Structure Connection Points moiety reference InChI atom numbers in the intermediate structure (the preceding molfile under the parent XML element). The first bond/positionNumber is again the ordinal number of the connection points moiety to be perfectly consistent with how this is done for proteins, nucleic acids and random polymers. The remaining bond/positionNumbers are the InChI atom numbers of the atoms in the distalMoiety structure which correspond to the connection points atoms in the intermediate structure. It is a kind of atom mapping, but not by enumerating all atoms but using an algorithm that allows a concise specification of a minimal set of atoms which "cut" a "hole" into the original structure into which the substituent structure is then inserted. That is why we call them "cut point atoms".



A minimal set of cut point atoms are specified as follows: In this example, 2, 7, 8, and 9 are specified and identify the moiety highlighted in green. Number 8 being the C of the nucleophile attack, 9 the Cl, 7 the entrance to the ring and 2 being any other member of this ring. The meaning is as follows: we determine the paths between all cut-point atoms, the members of the moiety are along these paths between the cut points. However, at every cut-point only those paths that link to another cut-point will be included. With cyclic structures we also need to constrain that not any path will be accepted, but only the shortest path; and finally, that paths are included in the order of the cut-points specified, in such a way that the next path added is the one which has the least number of "forbidden bonds". A "forbidden bond" is one which is cut out and not included by the previous cut-point atoms. For example, in this intermediate structure the bond from C to O is a forbidden bond because it was not already included by any path to another cut-point atom. This is the most parsimonious way to delineate moieties in any molecular structures by the smallest number of atoms, and this system can be canonicalized, for example, atom 4 or 3 could have been specified instead of 2 to "cut out" the same moiety, but 2 is the lowest InChI atom number so 2 is chosen.

This approach establishes a parsimonious minimal atom mapping, and all other atom mappings can be computed from there. This is similar to the fact that when we have mapped the atoms of the reaction center, then the other atoms mappings farther away from the center all follow.

Reduction into Practice: Reaction Patents

Until here we have presented how R-SPL can represent molecular reaction schemas. However, as the name "processStep" implies, the original design purpose of this feature was really describing chemical (pharmaceutical) processing steps that reduce the reaction from a general idea into the practice of synthetic chemistry. The realization here is that a reaction ultimately can be performed by adding enough practical details so that the reaction can be performed. Such details are determining suitable solvents and catalysts, reaction conditions (temperature, pressure), as well as preparatory steps and work-up steps. The schematic reaction will still be there in principle, but other steps and conditions will now be added.

A prominent type of document in chemistry is reaction patents for which file reaction-patent.xml (in the supplementary material ZIP file) is an example. This R-SPL file describes the patent WO 2010/027150 A2, claiming a new preparation of hydroxychloroquine [21]. All the parts of the patent (Abstract, Background Art, Claims etc.) are included as structured document text, with section and paragraph and table and figure entries, showing how R-SPL is indeed a document format which can be used to produce human readable documents. Reactions are presented using the same principle <processStep> and <interactor> elements already shown above. Only that now we are adding processStep codes which say

what is being done other than what reacts. A common action is stirring, such as in the instruction text, "`stir at 100 to 110℃ for 4 hours.`" This becomes:

```
<processStep>
  <code code="stir" codeSystem="1.3.6.1.4.1.32366.1.1.997" displayName="stir"/>
  <text>stir at 100 to 110℃ for 4 hours</text>
  <effectiveTime>
    <width value="4" unit="h"/>
  </effectiveTime>
  <controlVariable>
    <observation>
      <code code="temperature" codeSystem="1.3.6.1.4.1.32366.1.1.998"
            displayName="temperature"/>
      <value xsi:type="IVL_PQ">
        <low value="100" unit="Cel"/>
        <high value="110" unit="Cel"/>
      </value>
    </observation>
  </controlVariable>
</processStep>
```

Firstly we introduced ad-hoc codes such as "stir" and "temperature" where we use a meaningful word itself as the code. These codes could of course be sourced from an appropriate terminology or ontology once they are well defined there. But until such a terminology of chemical processes exists, we can simply make up codes and gather lists of codes somewhere.

Then this example shows how R-SPL represents activity parameters. Every action can have a time which may be a quite complex data element (e.g. for repeating activities), but in this case it is only specified as the duration of this stirring action, so the interval that would in a real performed activity have a start and end point in time, here it has only a width of 4 hours. Then there can be any number of other parameters, of which all settings, presets, we define as "control variables"

Reactions as Laboratory/Manufacturing Processes – Reaction SPL for Automated Synthesis

Patents provide some details as to how to reduce molecular reaction schemas into practice, but even patents are still schematic, disclosing only the most critical details which a person trained in the arts of synthetic chemistry can then fill in what is not specified to reproduce the reaction for themselves. But much more needs to be filled in. A good test for whether enough detail can be presented to allow for the reaction description to be actually performed is when we can control robots with them or an automated synthesis apparatus.

Automated synthesis and the tools to control synthetic robots is a cutting-edge topic in synthetic chemistry and drug design. Reaction SPL can represent such instructions. Autoprotocol [22] and XDL [23], [24] are recent formats for expressing actions that are compatible with synthesis robot systems. XDL is based on an abstraction of chemical assembly that enables one to create a state machine that can make arbitrary molecules. We use XDL as requirements specification for what the reaction SPL should support at a minimum. However, we also support specification of important properties such as melting point, boiling point, crystal morphology, natural state, viscosity, etc. of the participants, plus qualifying attributes such as concentration, diluent, chemical purity, enantiomeric excess etc.

We immediately find a large commonality in SPL already, as it had been designed to provide specific process steps of pharmaceutical manufacturing. Reaction SPL uses the process steps to represent chemical reaction schemas on a molecular level, but the original processStep features exist to describe reactions reduced into practice of synthetic processes.

We are providing a tabular overview (Table 2 in supplementary material ZIP file) to document the complete mapping of how the steps of the practical execution of a reaction in XDL correspond to Reaction SPL. Essentially XDL allows sequences of "steps" directly mapped to the SPL "processStep" element. In the XDL XML format, the process step names are the XML element names such as: Add, Transfer, FilterThrough, Stir, HeatChill, Dissolve, CleanVessel, Precipitate, Crystallize, EvacuateAndRefill, Purge, Filter, WashSolid, Dry, Separate, Evaporate, and Irradiate.

In SPL all process steps have the same element name “processStep” and what the step does is coded in the code sub-element. We can just use the XDL step names as processStep/codes. But our processStep/code also allows us to refer to other, wider, and especially more refined terminology of process steps (possibly aligned with domain specific ontologies developed by other groups).

XDL then has a set of parameters defined specifically for each step. For example the parameters “volume” and “amount” and “time” is defined for most steps. The “flow_rate” is only defined for Purge. “vessel” is defined for most steps and “from_vessel” and “to_vessel” defined for the Transfer step. Some parameters are Boolean (true/false) values, such as “viscous” for Add and Transfer, or “active” for HeatChill (to say if the temperature change should be effected by active heating or chilling or just by waiting for room temperature. These slight variances of meaning we address by specialized codes, instead of these parameters we say “Transfer-Viscous” or “HeatChill-Active” (and possibly even separate the step codes “HeatChill” into “Heat” vs. “Chill”.) Some parameters have small value sets of strings, such as “Separate” has “product_phase” with “top” vs. “bottom”. All such small differences can be encoded in the R-SPL processStep/ code.

We also note that XDL steps can be analyzed further to find that there are certain overlaps of meaning. For example: Add vs. Transfer, really have the same meaning: Add just Transfers a substance from a source vessel to a reactor, while Transfer might move an intermediary product out of one reaction vessel to some other reaction vessel. Ultimately, it is all the Transferring from one vessel to another. We could therefore just use the same code for it, or we define Add as a specialization of Transfer.

We also noted that there are some component actions implied by other steps. For example, the Dissolve step may have a stir speed parameter implying that the Dissolve step is practically just a kind of Stir step, with the only difference being the purpose of the stirring and the final objective, when the Dissolve step is done (i.e., when a clear solution has been produced without precipitate (or with precipitate if the intention is to create a saturated solution.) Likewise, the Separate step with its to_vessel parameter implies Transfer of the product phase into the receiving vessel. The same goes for the Filter step. Note even some reactant properties imply actions, such as a “stir” property and a “last_minute_addition”.

In the SPL processStep model (and generally the RIM Act model), all steps may have sub-steps. In fact, as a “rule of infinite decomposability”, every step can be decomposed into sub-steps as some more detail may always be required for some special use cases. There are two kinds of decomposition of steps into sub-steps, one is sequential, such as the Separate step being decomposed of four sub-steps: (1) addition of chloroform to a product mix of a previous reaction, (2) a time of stirring, (3) a time of waiting to allow the separation to occur, and finally (4) the transfer of the product phase into the receiving vessel. Defining these as a composite step with sub-steps is useful, because we can give a repeatNumber to the composite step, for example, to perform that separation step multiple times. In such a case, the separation may also include an evaporation step to remove the product phase solvent.

Other sub-actions occur in parallel, such as, for example, the Stir action which in XDL is implied for many other steps, such as Dissolve, Add, HeatChill, and many more. Active heating or chilling may accompany a Dissolve or other reaction action. In the SPL processStep model, parallel activities can be specified as component activities. The difference between sequential and parallel steps is indicated simply by a sequence number. Those processStep components with sequenceNumber 1, 2, 3, 4, … are sequential steps, when more than one processStep component has the same sequenceNumber, they are occurring in parallel.

There are split-codes (“split” is a term known from Workflow process specifications) that tell whether steps of same sequenceNumber occur in parallel or if only one of several steps is chosen based on some condition (like an if … then … else statement.) Because of the ability to provide conditional branches, our SPL / RIM Act model is like a Touring-complete language. Conditionals are specified with criteria observations. For example, we can say “the action Stir should be ongoing until the final-objective has been reached, identified by a “turbidity” measurement to reach the value a nominal value of “clear”. Or the Chilling can occur to keep a maintenance-objective as “temperature” between 20 and 40 °C.

There are also join-codes which indicate whether a parallel branch should follow the other parallel branches and be terminated when the other parallel branch(es) terminate, or whether it should be detached and continue even when the other parallel branch already terminated. An example for this is stirring during

dissolution. The dissolution step may have arrived at its goal, which then may stop the stirring, or in another setup the stirring should continue even as the transfer of the solution to another vessel is occurring.

In summary, the SPL processStep model has many more features than the XDL step model. In SPL process steps may be specified more explicitly, but ultimately it is simpler to use because instead of many parameter names and implicit process steps, everything can be stated with fewer features that can be used as in a programming language.

Fully Annotated Reaction with Side Products

Although often overlooked, particularly in the description of organic chemical transformations, the description of fully balanced reactions is an important feature when exploring the development of more intelligent synthetic design algorithms. Stoichiometrically accurate description of reactions lead to better insight into proper reaction controls (e.g. temperature, mixing) and monitoring and most importantly in post-reaction processing to isolate the product of interest. In the example shown here in the R-SPL file reaction-fully-annotated.xml provided in the supplementary material ZIP file, an amide is formed through the use of an activating agent (1-Cyano-2-ethoxy-2-oxoethylidenaminooxy)dimethylamino-morpholino-carbenium hexafluorophosphate (COMU) [25], which reacts with the acid to form an activated ester that then readily reacts with the aromatic amine moiety to form the product indicated. As a consequence of the activated ester formation the urea byproduct N,N-dimethylmorpholine-4-carboxamide is formed followed by the hydroxyl amine moiety ethyl cyanohydroxyiminoacetate (oxyma) upon reaction with the amine. In addition, the hexafluorophonic acid byproduct is trapped as the triethylamine salt.

In a molecular reaction schema, by-products are often not included because we only think about the desired reaction. Currently we simply list the by-products as products with a function code of “by-product” or “impurity.” We can provide approximate quantities with the reaction participants where one may at least describe order of magnitude relations. Creating fully consistent protocols that take into consideration the removal of such by-products which could include incompletely reacted substrates becomes exceeding important in defining product isolation protocols in an automated synthesis context. [26]

An aspect of virtually all reactions performed in the lab is the characterization of the products using analytical chemistry procedures, which today is usually some kind of chromatography, spectrometry, or NMR techniques. These characteristics observations can also be specified with the R-SPL files because the R-SPL files are built based on the SPL schema that was designed with Product Quality and Chemical Manufacturing Control (PQ/CMC) requirements. This means, a product is defined as having specifications, a set of observations and their result ranges that the product should conform to (e.g., the major peak in the LCMS and the maximum size of the minor peaks, along with characterization of the substance carried by those minor peaks). Then whenever the reaction is actually performed in a batch, the batch can be tested against that specification to find that it meets the specification.

Finally, when performing synthesis, whether manually or automated, we are dealing with more than abstract substances, in fact, the chemistry hardware does not really know or care about molecular structures, but they do very much care about the form and other characteristics of the actual material used for the reactants. E.g., is it a powder, and if yes, how fine? Is it a liquid? What is its viscosity? How should substances be mixed or added together given their properties? What is their solubility in different solvents? What can we used to wash our vessels? Some of these considerations are included in XDL already, but since XDL is designed for a type of reaction apparatus which carries most reagents as liquids, either a fluid or a suspension through pipes, pumps, and valves, it is limited in this regard.

Implementation

Standards, Implementation Guides, and Validation Rules

The current SPL standard and XML schema is based on SPL release 8. An upgraded release 9 had been prepared several years ago, which was specifically enhanced to provide full support for PQ/CMC requirements. For various reasons this had not been moved forward, because the only people with the requirement suddenly decided to do everything in a completely different way. No PQ/CMC standard had

been moved to any sort of implementation for many years since that decision was made, and whatever had been discussed for an alternative standard is far from capable of dealing with the chemical phenomena the way that SPL does. Now, what this means for R-SPL is that most of the use cases we have shown so far in this paper are handled by the current SPL release 8 schema, but we will release a draft for trial use (in one form or the other) of SPL release 9 with the PQ/CMC enhancements, and possibly with a few more edits found useful for full support of all advanced reaction SPL requirements.

Most people will find that the actual standard documentation itself is not very useful, since unfortunately the design of HL7 version 3 was, while conceptually very nimble and flexible, very constrained as to the expressivity of the presentation of the standard specification. In an over-zealous effort to make the standard specification formal and validated, the specification was released in diagrams and tables and databases and countless linked HTML pages but the documentation was not very user friendly. This is in large parts why the HL7 version 3 line of standards had collected some bad reputation and why HL7 itself had decided to start over from scratch, throwing away the superior model and going back to ad-hoc defined record formats where ad hoc data fields would be provided without systematic coherence. This may be an arguable statement, but the proof of this statement is that no useful PQ/CMC specification has been released let alone adopted which would even come close to being able to deal with molecular structures, reactions and process automation the way the SPL standard does, because these use cases are sacrificed on the altar of expediency.

Fortunately, the vast experience with implementing SPL in the pharmaceutical industry over more than a decade has been founded not on the complicated HL7 version 3 documentation as much as on quite a simple Implementation Guide document. That document is walking the implementer through by examples and “XML snippets”, i.e., parts of XML which implementers could copy and paste and edit the data and connect with other snippets that implementation becomes a very practically focused effort. Especially the production of SPL files is not a hard endeavor. The consumption of SPL files may be more involved if the objective is to be able to understand all types of SPL files. However, this is often not necessary, because fortunately SPL files are conceived of as documents so that all formal data elements can have free text presentations including pictures, diagrams, tables and other media. The implementation guide of basic SPL, with substance indexing and the new reaction SPL sections is being made available at https://www.chemspl.org/.

The power of the SPL Implementation Guide has come to a great extent from the fact that it has built in validation rules (or “validation procedures”), which are testable formal statements written over the XML content using the Schematron framework. I.e., XPath terms are written that either must produce a result (assert rules) or must not produce a result (report rules). We have been able to write rules that can reach into the depth of the chemical models, such as verifying that amino acid connection points actually reach N atoms that are an amino-group and C-atoms that are in a carboxyl group, same for phosphates, thio-phosphates and others, which is supported by a QuInChI implementation, a query extension of InChI [27] that is available for the Schematron (XPath) language. An R-SPL validator is also available at the https://www.chemspl.org/.

Implementation in Software

The chemoinformatics toolkit CACTVS [28], [29] has capabilities of reading and writing Substance Indexing SPL files for small molecule drugs. Capabilities to write and read Reaction SPL files are being implemented, as is an extension of the reaction object to store and process the full range of information which can be expressed in Reaction SPL. The current reaction object is limited to multi-step reaction data as found in MDL RXN/RDF, Reaction SMILES and similar formats. This simple data model will involve significant enhancements.

Pragmatic Data [30] has general HL7 RIM and SPL software that is used to process substance indexing files and substance (and then reaction databases) built on the general RIM model, so that very minimal software enhancements are required to create a reaction data bank. Many SPL implementations exist in the pharmaceutical sector, so this is something that not just a few individuals have made to work. In general since most of the chemical content is still carried in molfiles and InChIs and then references to atom numbers in these representations that are well known to chemoinformatics, the hurdle is not too high for many implementations to exist. No tall implementations need to support all possible features either. It is

perfectly OK to produce a limited implementation that mainly produces a certain subset of R-SPL documents and can only process certain features of such documents. This is the benefit of working with documents in the first place, i.e., that the data elements are there for when a consuming system sees value in using them, but they are not forced on every system that has no use for them, because humans can always go back to the text.

Discussion and Conclusions

We are proposing, and presenting a first version of, a new standard for reaction and reaction-related data description based on the Health Level Seven Structured Product Labeling data exchange standard. We have shown that R-SPL can seamlessly combine robust and reliable description of structural data and physical quantities in a machine-readable and fully semantic format using strict coded terminologies and quantitative parameters, with inclusion of not (yet) semantically resolved textual data in free-text paragraphs. Data schemas or models and databases are important for automated analysis, querying, and gathering inferences, but human language has infinite expressiveness. Humans like documents with their freedom of unconstrained expression. Data are best carried in the documents where they originate because information extraction and data mining are hard and error prone. It is a chore for people to have to enter data into computer systems; and data entry to computer systems divorces data from the original source. Databases with "comment" fields are not as useful because the original train of thought is butchered. Rich text document support is important for expressivity. Data should not be divorced from text. If you have a downstream system that requires data alone to be shared, and cannot handle the document text, then that text can simply be excluded for that system, but the users should not be deprived of a place to express themselves completely. Reaction SPL therefore allows the user to combine the best of both worlds.

Numerous other types of reaction-type documents and data sets can be expressed in R-SPL but were not included for this paper due to resource limitations. We mention here the possibility to encode reaction transforms such as SMIRKS, or the LHASA type transforms used in SAVI [31], as well as electronic lab notebook (ELN) records stored in repositories such as Chemotion [32]. Also, expressing tautomeric interconversion reactions is no problem in R-SPL due to the use of molfiles in which full connectivity can be listed (including hydrogens if desired). This is in contrast to RInChI [33], which suffers from the problem that InChI may be the same for at least some tautomeric pairs, in which case the reaction A ↔ A' becomes A ↔ A, i.e. no reaction is left.

Further developments of the Reaction SPL standard will be presented at https://www.chemspl.org/. We hope this new standard will provide a comprehensive way of representing and exchanging a broad range of reaction type data sets.

Disclaimer

The views and opinions presented here represent those of the authors and should not be considered to represent advice or guidance on behalf of the Food and Drug Administration. The content of this publication does not necessarily reflect the views or policies of the Department of Health and Human Services, nor does mention of trade names, commercial products or organizations imply endorsement by the US Government.

References

[1] M. C. Nicklaus, W. D. Ihlenfeldt, G. Blanke, P. N. Judson, and V. Delannée, "The Need for Comprehensive Reaction Handling in SAVI and Beyond," Noordwijkerhout,The Netherlands, 2018, vol. Program&Abstracts, p. 142 (P–62).

[2] "HL7 Standards Product Brief - HL7 Version 3: Reference Information Model (RIM) | HL7 International." https://www.hl7.org/implement/standards/product_brief.cfm?product_id=77 (accessed Oct. 05, 2021).

[3] "DailyMed - Download All Indexing & REMS Files." https://dailymed.nlm.nih.gov/dailymed/spl-resources-all-indexing-files.cfm (accessed Oct. 05, 2021).

[4] W. Warr, “National Institutes of Health (NIH) Workshop on Reaction Informatics,” Aug. 2021, doi: 10.33774/chemrxiv-2021-x5sj7.
[5] “NIH Virtual Workshop on Reaction Informatics, May 18-20, 2021.” https://cactus.nci.nih.gov/presentations/NIHReactInf_2021-05/NIHReactInf.html (accessed Oct. 05, 2021).
[6] G. Schadow, C. J. McDonald, J. G. Suico, U. Föhring, and T. Tolxdorff, “Units of Measure in Clinical Information Systems,” *Journal of the American Medical Informatics Association*, vol. 6, no. 2, pp. 151–162, Mar. 1999, doi: 10.1136/jamia.1999.0060151.
[7] S. Heller, A. McNaught, S. Stein, D. Tchekhovskoi, and I. Pletnev, “InChI - the worldwide chemical structure identifier standard,” *Journal of Cheminformatics*, vol. 5, no. 1, p. 7, Jan. 2013, doi: 10.1186/1758-2946-5-7.
[8] “Substance Indexing Files.” https://dailymed-data.nlm.nih.gov/public-release-files/substance_indexing_spl_files.zip (accessed Oct. 05, 2021).
[9] Y. Borodina and G. Schadow, “Representation of Proteins with Posttranslational Modifications in the HL7 SPL Standard,” Totowa, NJ: Humana Press, pp. 1–45. doi: 10.1007/7653_2018_31.
[10] H. Patel *et al.*, “SAVI, in silico generation of billions of easily synthesizable compounds through expert-system type rules,” *Sci Data*, vol. 7, no. 1, p. 384, Nov. 2020, doi: 10.1038/s41597-020-00727-4.
[11] “Synthetically Accessible Virtual Inventory (SAVI) Database Download Page.” https://cactus.nci.nih.gov/download/savi_download/ (accessed Jul. 03, 2017).
[12] “CTfile Formats (PDF) - Biovia Databases 2020.” Biovia, 2020. [Online]. Available: https://discover.3ds.com/sites/default/files/2020-08/biovia_ctfileformats_2020.pdf
[13] W. J. Cook, O. Senkovich, A. Hernandez, H. Speed, and D. Chattopadhyay, “Biochemical and structural characterization of Cryptosporidium parvum Lactate dehydrogenase,” *International Journal of Biological Macromolecules*, vol. 74, pp. 608–619, Mar. 2015, doi: 10.1016/j.ijbiomac.2014.12.019.
[14] “Jmol: an open-source Java viewer for chemical structures in 3D.” http://jmol.sourceforge.net/ (accessed Oct. 22, 2021).
[15] S. Fujita, “Description of organic reactions based on imaginary transition structures. 1. Introduction of new concepts,” *J. Chem. Inf. Comput. Sci.*, vol. 26, no. 4, pp. 205–212, Nov. 1986, doi: 10.1021/ci00052a009.
[16] F. Hoonakker, N. Lachiche, A. Varnek, and A. Wagner, “Condensed Graph of Reaction: Considering a Chemical Reaction As One Single Pseudo Molecule,” *Springer*, Jul. 2009, [Online]. Available: http://dtai.cs.kuleuven.be/ilp-mlg-srl/papers/ILP09-5.pdf
[17] A. Varnek, D. Fourches, F. Hoonakker, and V. P. Solov’ev, “Substructural fragments: an universal language to encode reactions, molecular and supramolecular structures,” *J Comput Aided Mol Des*, vol. 19, no. 9, pp. 693–703, Sep. 2005, doi: 10.1007/s10822-005-9008-0.
[18] V. Delannée and M. C. Nicklaus, “ReactionCode: format for reaction searching, analysis, classification, transform, and encoding/decoding,” *J Cheminform*, vol. 12, no. 1, p. 72, Dec. 2020, doi: 10.1186/s13321-020-00476-x.
[19] “Rethinking the Chemical Reaction as a Graph: Imaginary Transition Structures and Beyond.” http://depth-first.com/articles/2020/02/24/rethinking-the-chemical-reaction-as-a-graph-imaginary-transition-structures-and-beyond/ (accessed Oct. 13, 2021).
[20] “Walkthrough of Substitution Reactions (1) - Introduction,” *Master Organic Chemistry*, May 31, 2012. https://www.masterorganicchemistry.com/2012/05/31/walkthrough-of-substitution-reactions-1-introduction/ (accessed Oct. 13, 2021).
[21] Y. S. Min, H.-S. Cho, and K. W. Mo, “New Preparation of Hydroxychloroquine,” Mar. 11, 2010 Accessed: Oct. 13, 2021. [Online]. Available: https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2010027150
[22] “Autoprotocol.” https://autoprotocol.org/ (accessed Oct. 22, 2021).
[23] “XDL 1 Standard — xdl 0.5.0 documentation.” https://croningroup.gitlab.io/chemputer/xdl/standard/index.html (accessed Oct. 05, 2021).
[24] “Cronin Group / Chemputer / XDL,” *GitLab*. https://gitlab.com/croningroup/chemputer/xdl (accessed Oct. 05, 2021).
[25] A. El-Faham, R. S. Funosas, R. Prohens, and F. Albericio, “COMU: A Safer and More Effective Replacement for Benzotriazole-Based Uronium Coupling Reagents,” *Chemistry – A European Journal*, vol. 15, no. 37, pp. 9404–9416, 2009, doi: 10.1002/chem.200900615.

[26] A. G. Godfrey, T. Masquelin, and H. Hemmerle, “A remote-controlled adaptive medchem lab: an innovative approach to enable drug discovery in the 21st Century,” *Drug Discovery Today*, vol. 18, no. 17, pp. 795–802, Sep. 2013, doi: 10.1016/j.drudis.2013.03.001.
[27] “QuinChi - A variation of InChI for expressing structure queries,” presented at the NIH Virtual Workshop on InChI, Mar. 23, 2021. [Online]. Available: https://cactus.nci.nih.gov/presentations/NIHInChI_2021-03/Day_2_QuInChI-talk.ppt
[28] W. Ihlenfeldt, Y. Takahashi, H. Abe, and S. Sasaki, “Computation and Management of Chemical-Properties in Cactvs - an Extensible Networked Approach Toward Modularity and Compatibility,” *J. Chem. Inf. Comput. Sci.*, vol. 34, no. 1, pp. 109–116, Feb. 1994, doi: 10.1021/ci00017a013.
[29] “CACTVS Documentation.” http://www.xemistry.com/docs.htm (accessed Oct. 31, 2013).
[30] “Pragmatic Data LLC - Expert in healthcare and pharmaceutical domain software development.” https://www.pragmaticdata.com/ (accessed Oct. 05, 2021).
[31] P. N. Judson, W.-D. Ihlenfeldt, H. Patel, V. Delannée, N. Tarasova, and M. C. Nicklaus, “Adapting CHMTRN (CHeMistry TRaNslator) for a New Use,” *J. Chem. Inf. Model.*, vol. 60, no. 7, pp. 3336–3341, Jul. 2020, doi: 10.1021/acs.jcim.0c00448.
[32] “Chemotion.” https://www.chemotion-repository.net/welcome (accessed Oct. 13, 2021).
[33] G. Grethe, G. Blanke, H. Kraut, and J. M. Goodman, “International chemical identifier for reactions (RInChI),” *Journal of Cheminformatics*, vol. 10, no. 1, p. 22, May 2018, doi: 10.1186/s13321-018-0277-8.