An alternative description of molecular states in XSAMS

Dr Christian Hill

Department of Physics and Astronomy

UCL

The current Schema

```
<ElectronicHome>
  <ElectronicComponent>
    <VibrationalHome>
      <VibrationalComponent>
        <VibrationalQuantumNumbers>
          <VibrationalNu>
            <Label>v1</Label>
            <Value>1</Value>
          </VibrationalNu>
          <VibrationalNu vibrationInversion="s">
            <Label>v2</Label>
            <Value>0</Value>
          </VibrationalNu>
          <VibrationalNu vibrationLNu-i="0">
            <Label>v3</Label>
            <Value>0</Value>
          </VibrationalNu>
          <VibrationalNu vibrationLNu-i="2">
            <Label>v4</Label>
            <Value>2</Value>
          </VibrationalNu>
        </VibrationalQuantumNumbers>
```

The J=22, K=10 state of the $(1,0^{+},0^{0},2^{2})$ vibrational state of NH_3

```
      <RotationalHome>
        <RotationalComponent>
          <NonLinearPolyatomic>
            <NonLinearNoElecNoHyperF>
              <TotalAngularMomentumN>
                <Label>J</Label>
                <Value>22</Value>
              </TotalAngularMomentumN>
              <MolecularProjection>
                <TotalMolecularProjectionN>
                  <Label>K</Label>
                  <Value>10</Value>
                </TotalMolecularProjectionN>
              </MolecularProjection>
            </NonLinearNoElecNoHyperF>
          </NonLinearPolyatomic>
        </RotationalComponent>
      </RotationalHome>
    </VibrationalComponent>
  </VibrationalHome>
</ElectronicComponent>
</ElectronicHome>
```

The current Schema

```
<ElectronicHome>
  <ElectronicComponent>
    <VibrationalHome>
      <VibrationalComponent>
        <VibrationalQuantumNumbers>
          <VibrationalNu>
            <Label>v1</Label>
            <Value>1</Value>
          </VibrationalNu>
          <VibrationalNu vibrationInversion="s">
            <Label>v2</Label>
            <Value>0</Value>
          </VibrationalNu>
          <VibrationalNu vibrationLNu-="0">
            <Label>v3</Label>
            <Value>0</Value>
          </VibrationalNu>
          <VibrationalNu vibrationLNu-i="2">
            <Label>v4</Label>
            <Value>2</Value>
          </VibrationalNu>
        </VibrationalQuantumNumbers>
```

The J=22, K=10 state of the $(1,0^{+},0^{0},2^{2})$ vibrational state of NH_3

```
        <RotationalHome>
          <RotationalComponent>
            <NonLinearPolyatomic>
              <NonLinearNoElecNoHyperF>
                <TotalAngularMomentumN>
                  <Label>J</Label>
                  <Value>22</Value>
                </TotalAngularMomentumN>
                <MolecularProjection>
                  <TotalMolecularProjectionN>
                    <Label>K</Label>
                    <Value>10</Value>
                  </TotalMolecularProjectionN>
                </MolecularProjection>
              </NonLinearNoElecNoHyperF>
            </NonLinearPolyatomic>
          </RotationalComponent>
        </RotationalHome>
      </VibrationalComponent>
    </VibrationalHome>
  </ElectronicComponent>
</ElectronicHome>
```

Advantages of the current Schema

- *Describes* the specific quantum numbers used to identify a state

e.g.

K is explicitly the TotalMolecularProjection of the TotalAngularMomentum of a NonLinearPolyatomic molecule with

NoElec[tronic and]NoHyperF[ine] angular momentum.

Advantages of the current Schema

- *Restricts* which quantum numbers can be used to describe a molecular state, and (to some extent) what values they can take

e.g.

The NonLinearNoElecNoHyperF molecules may have only angular momentum quantum numbers N (*i.e.* J) and K (and, maybe, M_J).

Disadvantages of the current Schema

- Verbose and complex
  - Slow (sometimes *very* slow) to create and parse
  - Large file sizes (which compress well)
  - Doesn't easily map to relational database tables (for searching, storing and maintaining data)
  - Doesn't easily map to ASCII text tables (the output format that most users likely want)

Disadvantages of the current Schema

- Hard to *compare* states
  - In general, a difficult problem anyway
  - Quantum numbers and symmetries present in many, dispersed XML *elements* and *attributes*
  - Possible approaches:
    - Parse the XML first – requires XSAMS-aware software
    - Compare the XML source directly - slow

Disadvantages of the current Schema

- Some useful information is omitted- for example:
  - No way to easily identify *which nucleus* is being coupled to for hyperfine quantum numbers
  - Vibrational normal modes are identified by arbitrary text labels:

```
<VibrationalNu>
  <Label>v1</Label>
  <Value>1</Value>
</VibrationalNu>
```

Disadvantages of the current Schema

- Doesn't actually *restrict* content that much – for example:
  - Inappropriate quantum numbers can easily be used: *e.g.* one can give NH_3 AsymmetricKa, AsymmetricKc and AsymmetricTau quantum numbers
  - "Unphysical" states (*e.g.* $J = 22$, $K = 23$) can easily be specifed

Disadvantages of the current Schema

- Difficult to extend to include new quantum numbers and symmetries:
  - Use `SerialQuantumNumber` for arbitrary content?
  - Who maintains the Schema?
  - For how long?
  - How are changes suggested and assessed?

An alternative approach

- Simplest – elements for just quantum numbers (QN) and symmetries (Sym), identified by a “name” attribute:

```
<QNs>
   <QN name="J">22</QN>
   <QN name="K">10</QN>
   <QN name="v1">1</QN>
   <QN name="v2">0</QN>
   <Sym name="vibInv">s</Sym>
   ...
</QNs>
```

The “case-by-case” approach

- “Case-by-case” – introduce restrictions and canonical names for quantum numbers in common cases: *e.g.* closed-shell diatomics:

```
<dcs:QNs>
   <dcs:ElecStateLabel>X</dcs:ElecStateLabel>
   <dcs:J>8</dcs:J>
   <dcs:v>1</dcs:v>
   <dcs:parity>+</dcs:parity>
</dcs:QNs>
```

where the prefix `dcs` refers to an *XML namespace*

The “case-by-case” approach

- Different states of the same molecule may belong to different cases:
  - H_2O: normal mode description *vs.* local mode
  - NO: Hund’s case (a) – case (b) transition
  - CO: closed-shell in its electronic ground-state, can describe excited electronic states with e.g. Hund’s case (a) / (b) quantum numbers

The “case-by-case” approach

- Another example: the $J = 22$, $K = 10$ state of the $(1,0^{+},0^{0},2^{2})$ vibrational state of NH_3 – a closed-shell symmetric top (`stcs`):

```
<stcs:QNs assigned="no hyperfine">
   <stcs:ElecStateLabel>X</stcs:ElecStateLabel>
   <stcs:J>20</stcs:J>
   <stcs:K>10</stcs:K>
   <stcs:vi mode="1">1</stcs:vi>
   <stcs:vi mode="2">0</stcs:vi>
   <stcs:vi mode="3">0</stcs:vi>
   <stcs:vi mode="4">2</stcs:vi>
   <stcs:li mode="2">0</stcs:li>
   <stcs:li mode="4">2</stcs:li>
   <stcs:vibInv>s</stcs:vibInv>
</stcs:QNs>
```

The “case-by-case” approach

- Identified cases with [examples]:
  - `dcs`: closed-shell diatomics [CO, N_2]
  - `hunda`: open-shell diatomics using Hund’s case (a) [OH]
  - `hundb`: open-shell diatomics using Hund’s case (b) [NO, O_2]
  - `stcs`: closed-shell symmetric tops [NH_3, CH_3Cl]
  - `lpcs`: closed-shell linear polyatomics [C_2H_2]
  - `lpos`: open-shell linear polyatomics [C_3H]
  - `nlpcs`: closed-shell non-linear polyatomics [H_2CO]

Advantages of the “case-by-case” approach

- Simpler, more concise, “flat” format
  - is easier and quicker to parse and write
  - generates smaller files
- Much more direct relationship to the relational database and text table formats
  - easier to compare molecular states
  - more likely to be adopted outside the VAMDC?

Advantages of the “case-by-case” approach

- Better control of allowed quantum numbers and symmetries for common case
  - harder to “misuse” XML tags
  - can control whether particular quantum numbers can be integral or half-integral

- Additional information (*e.g.* normal mode designation, coupled magnetic nuclei IDs) can be specified as attributes

Advantages of the “case-by-case” approach

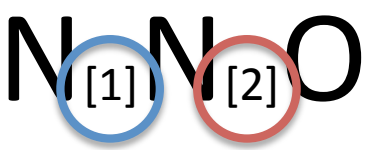
- Easier to extend
  - Data providers can provide a Schema with their data if necessary
- Simpler to specify unusual quantum numbers and symmetries with generic `QN` and `Sym` elements
  - A bit like `SerialQuantumNumber`, but in one place

Advantages of the “case-by-case” approach

- Better description of hyperfine coupling with the `nuclearSpinRef` attribute - *e.g.* N_2O):

```
<MoleculeNuclearSpins>
  <AtomArray>
    <AtomN elementSymbol="N" isotope="14”
          nuclearSpin="1.0" nuclearSpinID="Q_N1"/>
    <AtomN elementSymbol="N" isotope="14”
          nuclearSpin="1.0" nuclearSpinID="Q_N2"/>
    <AtomN elementSymbol="O" isotope="16”
          nuclearSpin="0.0" nuclearSpinID="Q_O"/>
  </AtomArray>
  <BondArray>
     <Bond atomRefs2="Q_N2 Q_O"/>
     <Bond atomRefs2="Q_N1 Q_N2"/>
  </BondArray>
</MoleculeNuclearSpins>
...
```

```
<ltcs:QNs assigned="true">
  <ltcs:ElecStateLabel>
    X
  </ltcs:ElecStateLabel>
  <ltcs:v1>0</ltcs:v1>
  ...
  <ltcs:F1 nuclearSpinRef="Q_N1">
    1
  </ltcs:F1>
  <ltcs:F nuclearSpinRef="Q_N2">
    2
  </ltcs:F>
</ltcs:QNs>
```



Disadvantages of the “case-by-case” approach

- As currently conceived, uses NVDL = Namespace-based Validation Dispatch Language, so validating the XML is more complex
- Needs a separate dictionary and associated software tools to look up cases
- Currently only contains limited information about the *electronic structure* of the molecule