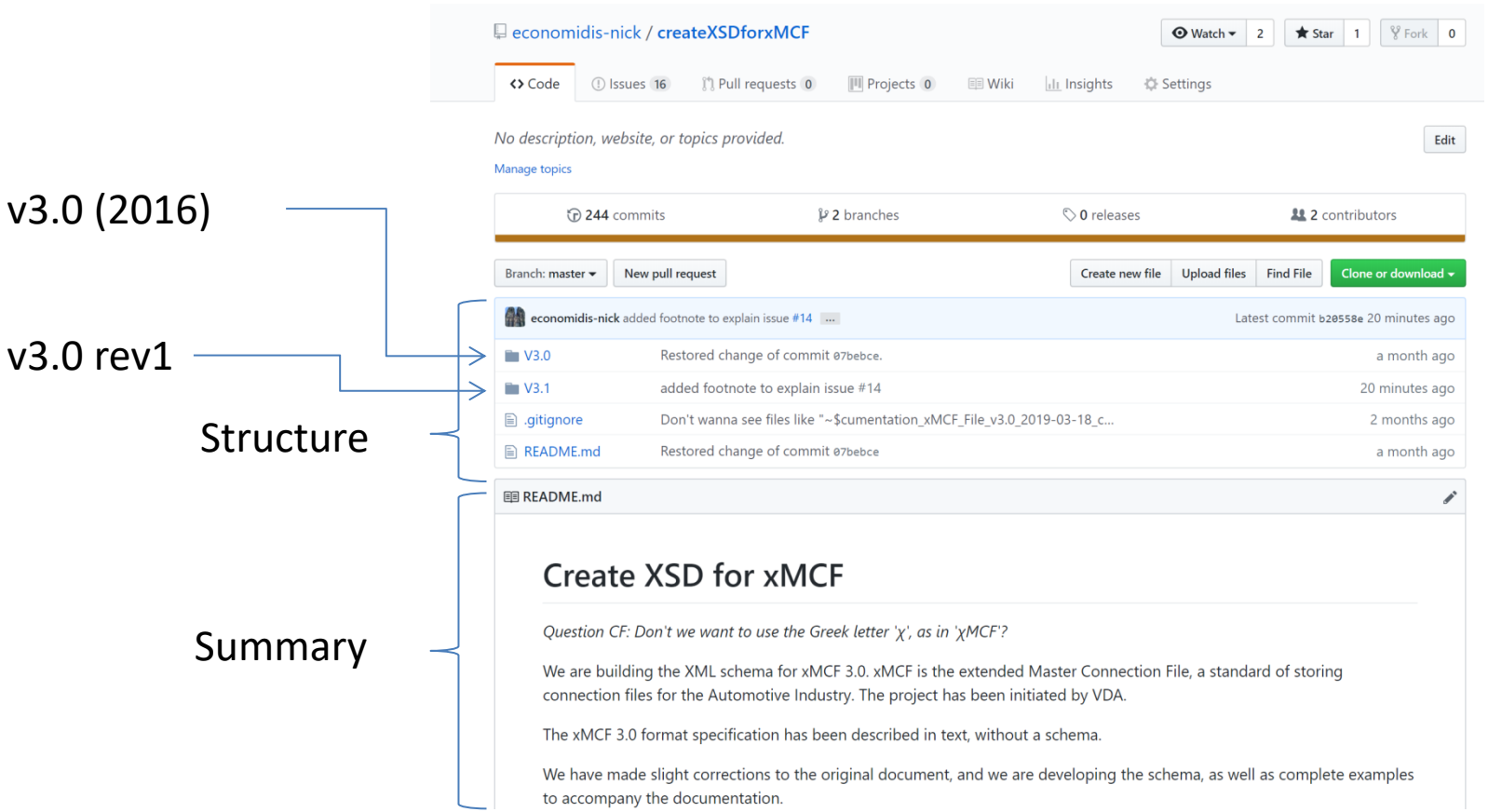
xMCF 3.0 schema and documentation revision 1

Nick Economidis, BETA-CAE Systems

How to access it on Github

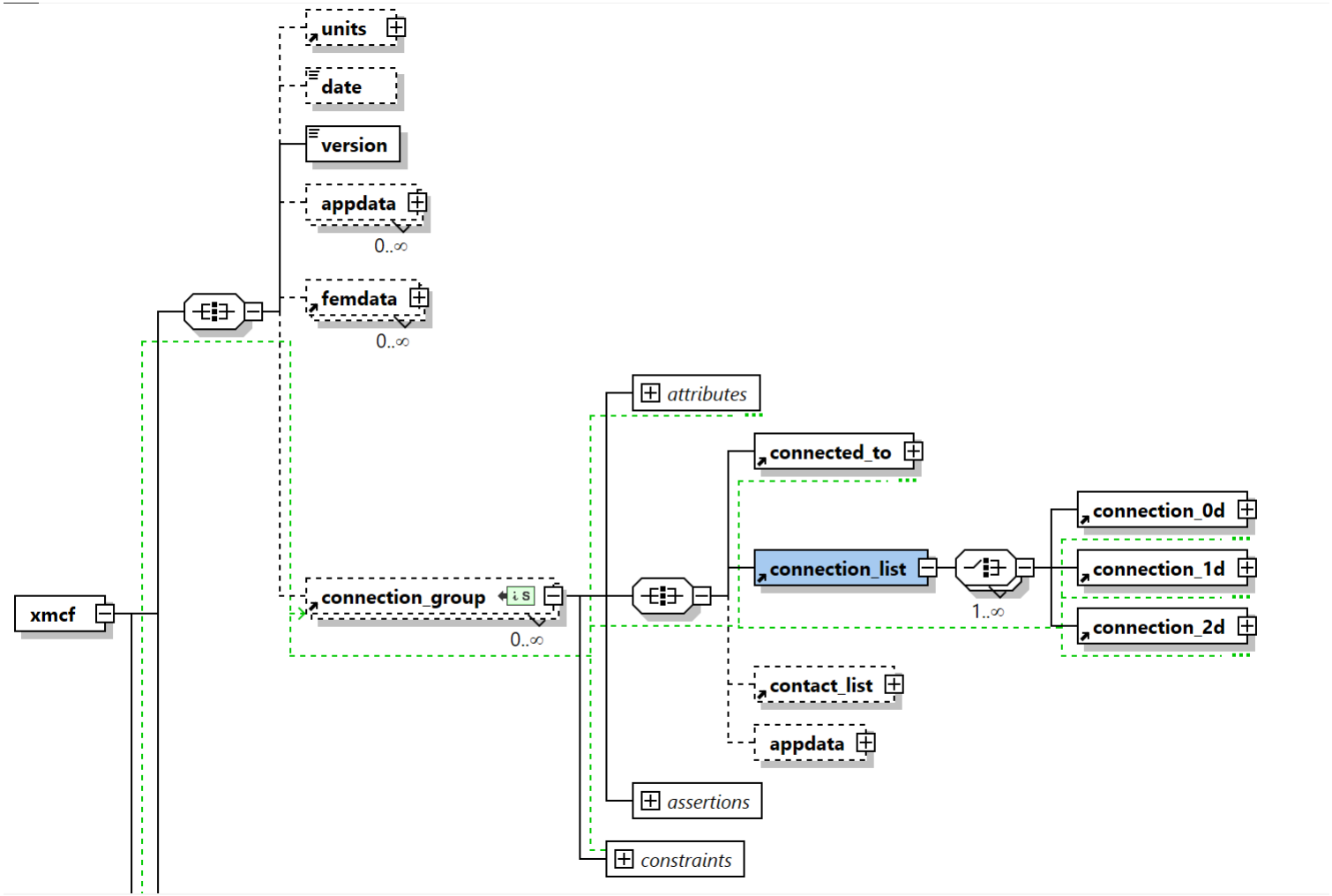
Hosted page:

https://github.com/economidis-nick/createXSDforxMCF



Database Schema (xsd)

- schema best browsed with XML Spy (demo)



Database Schema (xsd)

- assertions
  - dynamic conditions between 2 or more values

easy to read error message

condition

```
<xs:assert xerces:message="rivet: it must hold that (head_height + sink_size) > 0"
        test="
        if (exists(@head_height) and exists(@sink_size))
        then (sum(@head_height + @sink_size) gt 0)
        else true()"/>
<xs:assert xerces:message="rivet: it must hold that (head_height + sink_size) > 0"
        test="
        if (exists(@head_height) and not(exists(@sink_size)))
        then (@head_height gt 0)
        else true()"/>
<xs:assert xerces:message="rivet: it must hold that (head_height + sink_size) > 0"
        test="
        if (not(exists(@head_height)) and exists(@sink_size))
        then (@sink_size gt 0)
        else true()"/>
```

Database Schema (xsd)

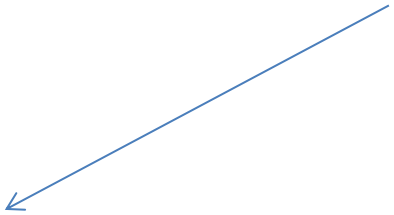
- comments
  - used to explain design decisions

```
<xs:complexType name="appdata_type">
        <!--
        NOTE: We chose not to restrict the choice of elements under appdata, because the spec says:
        "The following list does not imply that other application names are forbidden.
        Its noly purpose is to reserve the registered names against inappropriate use".
        -->
        <xs:sequence>
                <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
</xs:complexType>
```

Database Schema (xsd)

- intermediate data types
  - reduces copying
  - used the "**_type**" naming convention

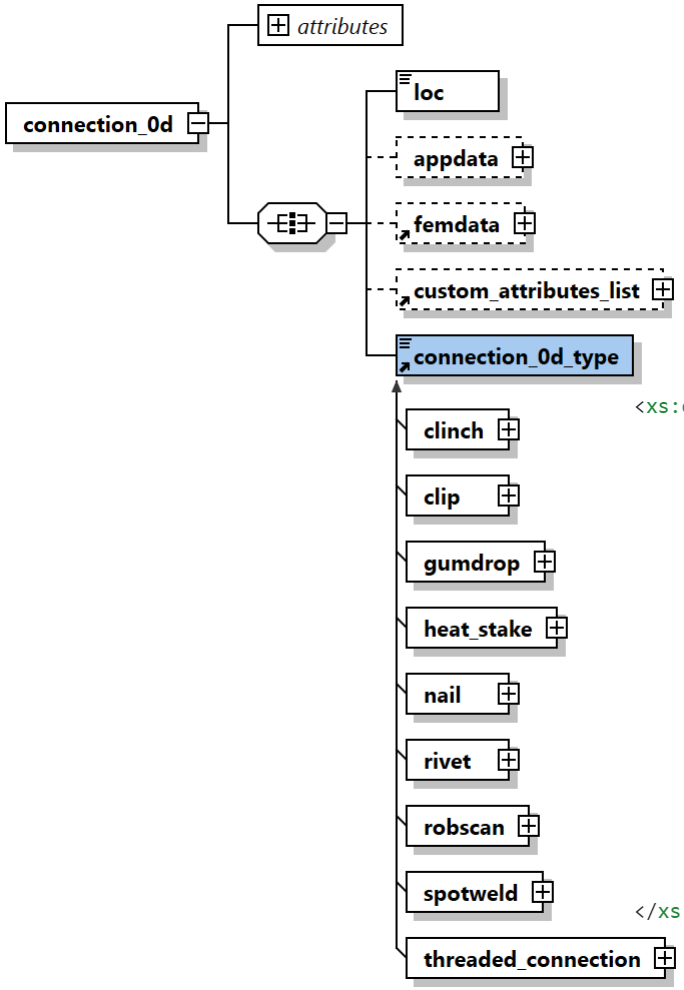
```
<xs:element name="femdata">
        <xs:complexType>
                <xs:all>
                        <xs:element name="PAMCRASH"   type="femdata_type" minOccurs="0" maxOccurs="1"/>
                        <xs:element name="LSDYNA"     type="femdata_type" minOccurs="0" maxOccurs="1"/>
                        <xs:element name="PERMAS"     type="femdata_type" minOccurs="0" maxOccurs="1"/>
                        <xs:element name="ABAQUS"     type="femdata_type" minOccurs="0" maxOccurs="1"/>
                        <xs:element name="RADIOSS"    type="femdata_type" minOccurs="0" maxOccurs="1"/>
                        <xs:element name="OPTISTRUCT" type="femdata_type" minOccurs="0" maxOccurs="1"/>
                        <xs:element name="NASTRAN"    type="femdata_type" minOccurs="0" maxOccurs="1"/>
                        <xs:element name="FEMFAT"     type="femdata_type" minOccurs="0" maxOccurs="1"/>
                </xs:all>
        </xs:complexType>
</xs:element>
```



```
<xs:complexType name="femdata_type">
        <xs:sequence>
                <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>
        </xs:sequence>
</xs:complexType>
```

Database Schema (xsd)

- aternative elements
  - **substitutionGroup** acts like “is-a relationship” for elements




```
<xs:element name="spotweld" substitutionGroup="connection_0d_type sequence_connection_0d_type">
        <xs:complexType>
                <xs:attribute name="diameter" type="floatPositive"/>
                <xs:attribute name="technology">
                        <xs:simpleType>
                                <xs:restriction base="xs:string">
                                        <xs:enumeration value="resistance"/>
                                        <xs:enumeration value="laser"/>
                                        <xs:enumeration value="projection"/>
                                        <xs:enumeration value="friction"/>
                                </xs:restriction>
                        </xs:simpleType>
                </xs:attribute>
        </xs:complexType>
</xs:element>
```

Database Schema (xsd)

- database relations
  - **primary / secondary keys**
    e.g. makes sure “`fixed_to`” is always one of “`connected_to/part/index`”

```
<xs:key name="connected_to_key">
        <xs:selector xpath="connected_to/*"/>
        <xs:field xpath="@index"/>
</xs:key>
```

```
<xs:keyref name="fixed_to_index" refer="connected_to_key">
        <xs:selector xpath="connection_list/connection_0d/threaded_connection/bolt"/>
        <xs:field xpath="@fixed_to"/>
</xs:keyref>
```