

# 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>

v3.0 (2016)

v3.0 rev1

Structure

Summary

economidis-nick / createXSDforxMCF

Watch 2 Star 1 Fork 0

Code Issues 16 Pull requests 0 Projects 0 Wiki Insights Settings

No description, website, or topics provided. Edit

Manage topics

244 commits 2 branches 0 releases 2 contributors

Branch: master New pull request Create new file Upload files Find File Clone or download

economidis-nick added footnote to explain issue #14 Latest commit b2e558e 20 minutes ago

V3.0	Restored change of commit e7bebc.	a month ago
V3.1	added footnote to explain issue #14	20 minutes ago
.gitignore	Don't wanna see files like "~-\$cumentation_xMCF_File_v3.0_2019-03-18_c...	2 months ago
README.md	Restored change of commit e7bebc	a month ago

README.md

## Create XSD for xMCF

Question CF: Don't we want to use the Greek letter 'χ', as in 'χMCF'?

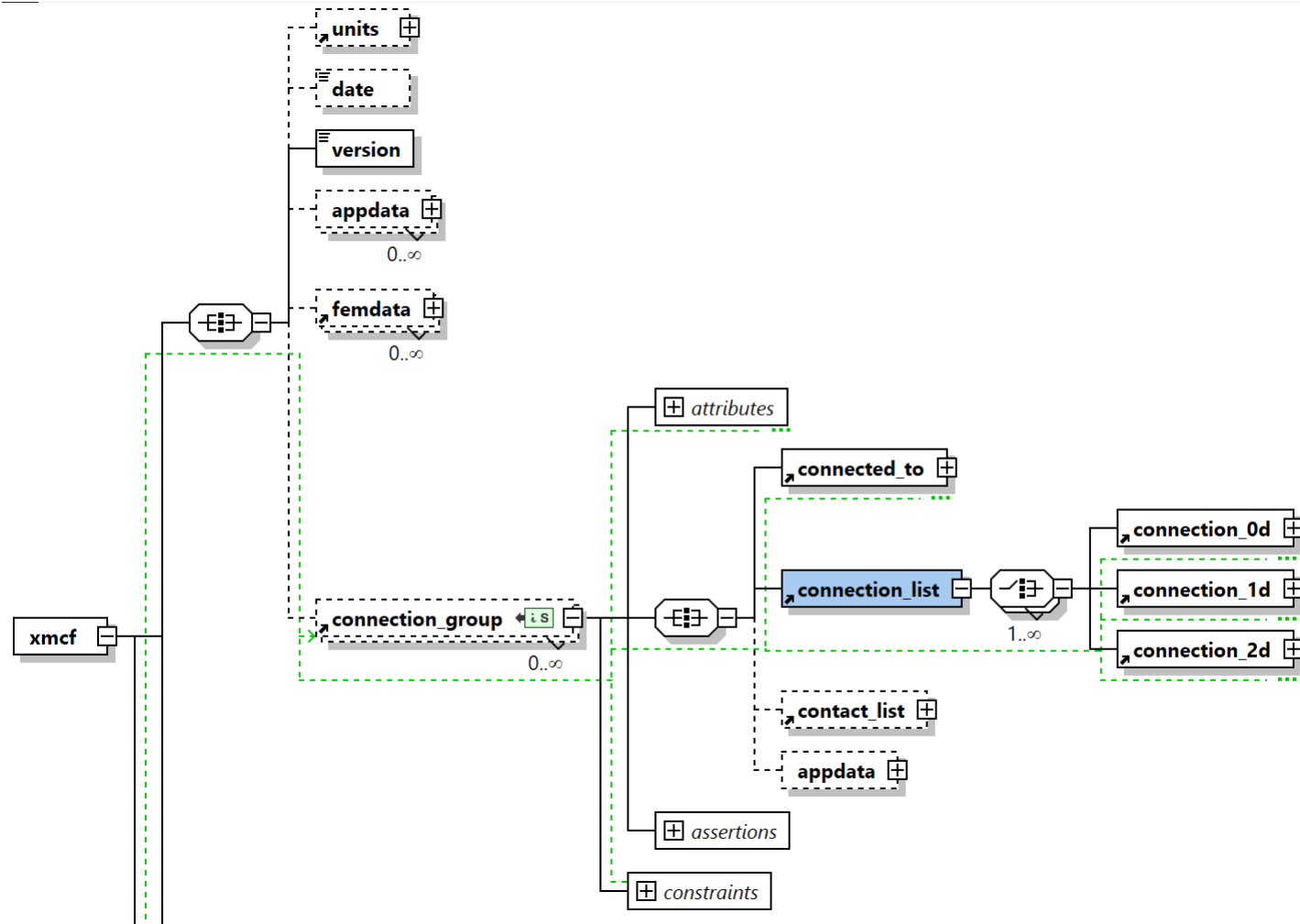
We are building the XML schema for xMCF 3.0. xMCF is the extended Master Connection File, a standard of storing connection files for the Automotive Industry. The project has been initiated by VDA.

The xMCF 3.0 format specification has been described in text, without a schema.

We have made slight corrections to the original document, and we are developing the schema, as well as complete examples to accompany the documentation.

# 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 only 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

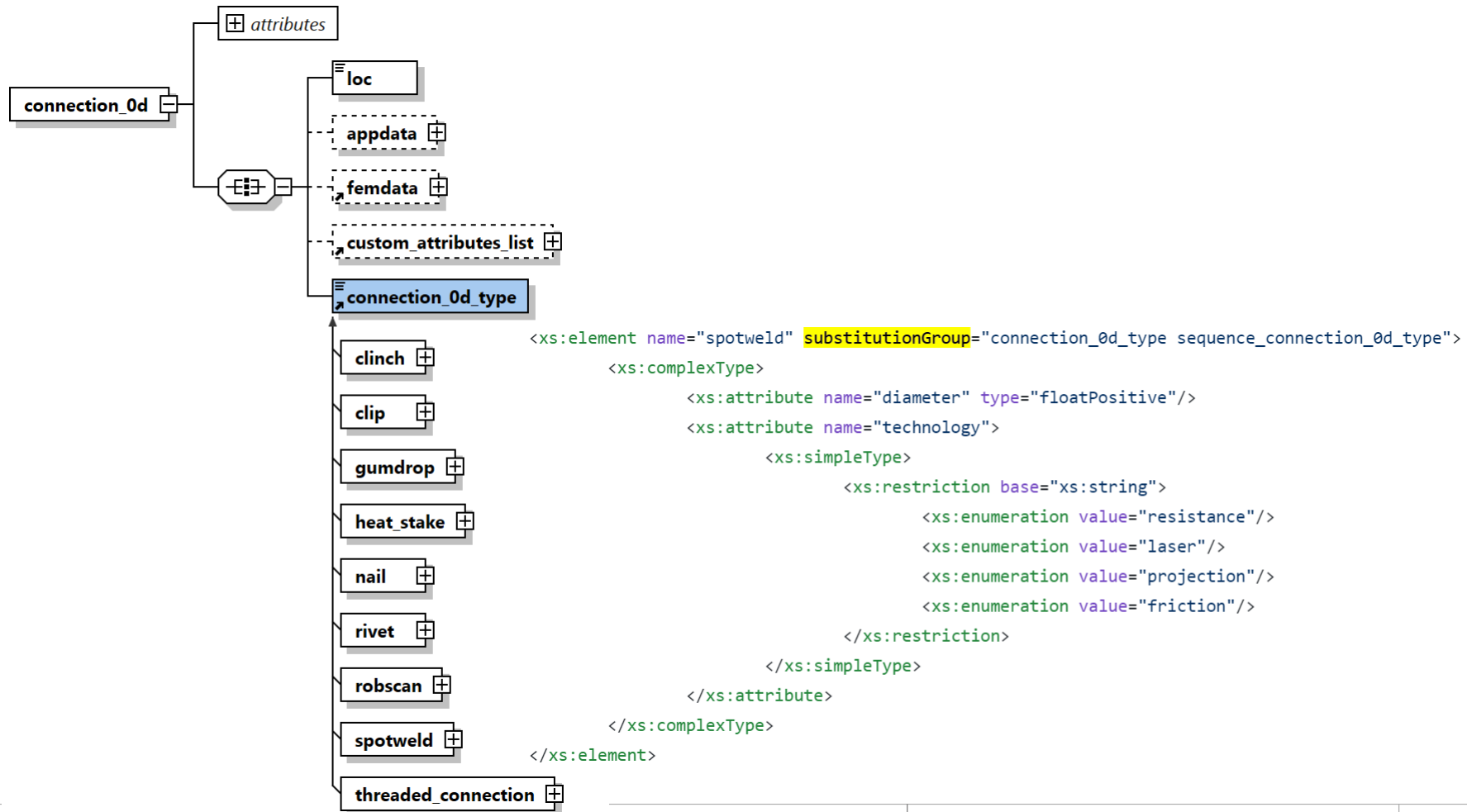
```
<xs:element name="femdata">
  <xs:complexType>
    <xs:all>
      <xs:element name="PAMCRASH" type="femdata_type" minOccurs="0" maxOccurs="1"/>
      <xs:element name="LSDYNA" type="femdata_type" minOccurs="0" maxOccurs="1"/>
      <xs:element name="PERMAS" type="femdata_type" minOccurs="0" maxOccurs="1"/>
      <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)

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

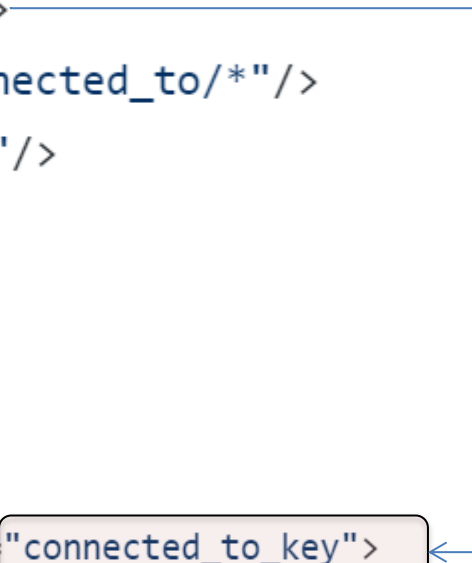


## Database Schema (xsd)

- 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>
```

A blue line with arrows at both ends connects the 'connected\_to\_key' attribute in the first code block to the 'refer="connected\_to\_key"' attribute in the second code block, illustrating the relationship between the key and the keyref.