

Type Substitution support in Microsoft InfoPath

Marc Gratacos 2005.01.24

FpML, the Financial products Markup Language, is an XML vocabulary focused on over-the-counter (OTC) financial instruments. These are financial products which are not traded via an exchange market, but directly between two financial institutions, such as banks. FpML is built using W3C Schema.

Recently, we have been working on developing forms using the FpML Schema.

We are pleased with the Schema capabilities of InfoPath but we have encountered one main issue, which made our development very difficult.

Support for Type Substitution across the entire Schema

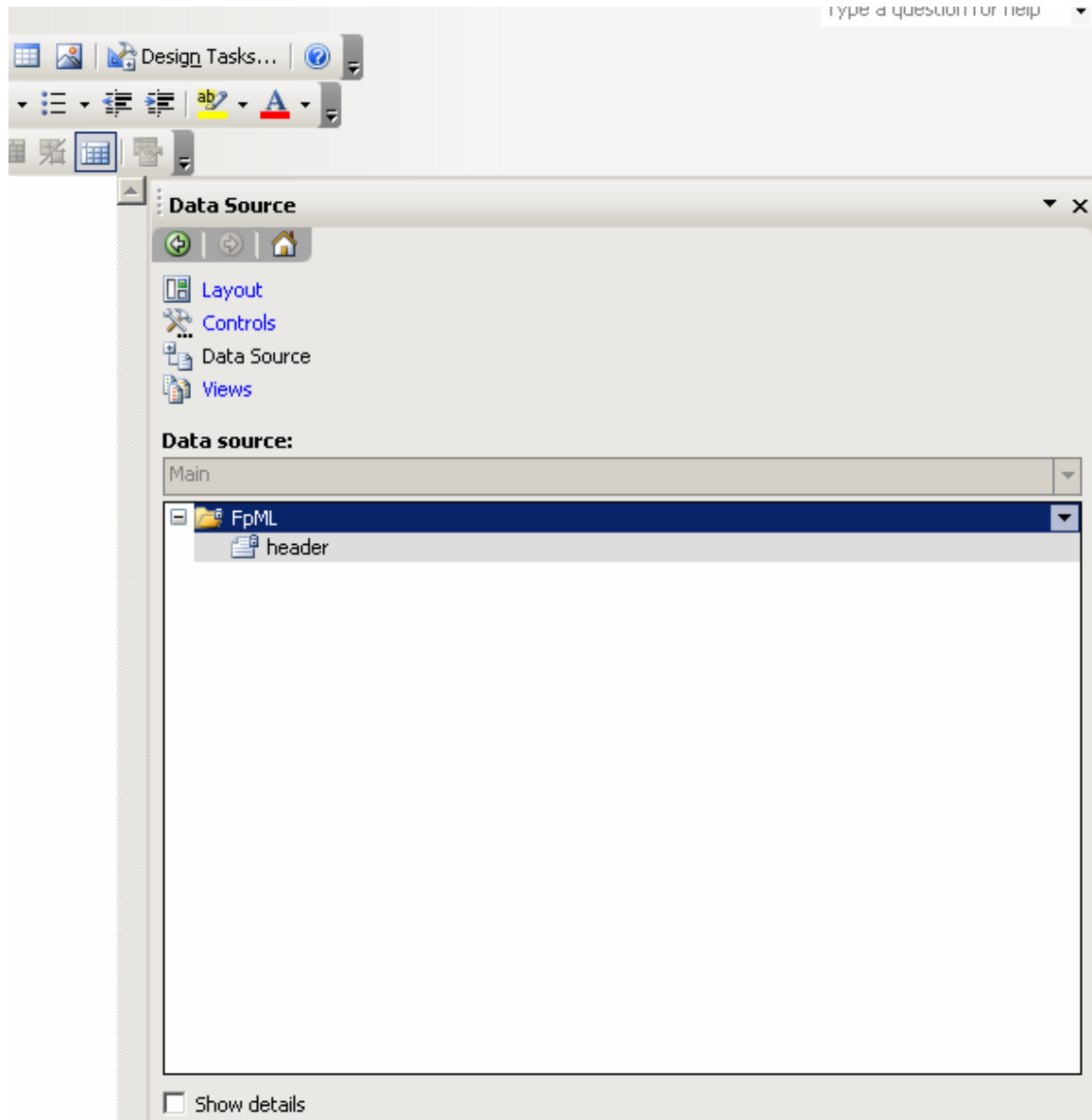
FpML uses extensively type substitution to determine different content models. I created a very simple example (Example 1) which shows what it is happening using FpML.

Example 1:

```
<?xml version="1.0" encoding="UTF-8"?>
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <xsd:element name="FpML" type="Document"/>
  <xsd:complexType name="Document" abstract="true"/>
    <xsd:complexType name="DataDocument">
      <xsd:complexContent>
        <xsd:extension base="Document">
          <xsd:sequence>
            <xsd:element name="product"
type="xsd:string"/>
          </xsd:sequence>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>
    <xsd:complexType name="Message">
      <xsd:complexContent>
        <xsd:extension base="Document">
          <xsd:sequence>
            <xsd:element name="header"
type="xsd:string"/>
          </xsd:sequence>
        </xsd:extension>
      </xsd:complexContent>
    </xsd:complexType>
  </xsd:schema>
```

There are two main content models in FpML. the one defining the DataDocument and the one defining the Message. This is done using type substitution. We would expect InfoPath to give us the ability to choose between the two content models. Arbitrarily, InfoPath is showing the content model of Message (see Image 1) but there are two possible content models: DataDocument and Message.

Image 1:



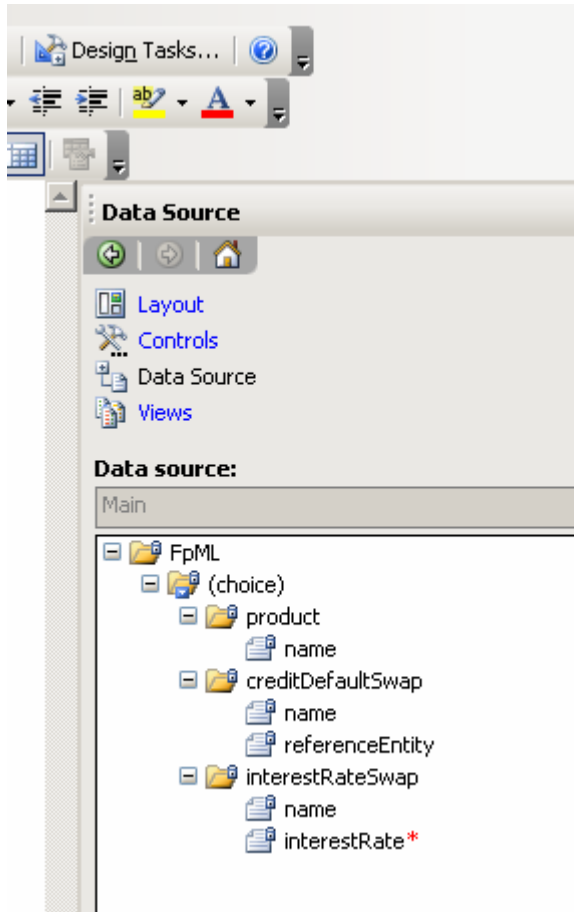
The first question is: **Why is InfoPath choosing arbitrarily one of the derived types (Message)?**

We would like to be able **to choose** the content model similarly to the behavior that we get with InfoPath using substitution groups, where there is a 'choice' control with the ability to choose between the substitute elements. For example, we have the following Schema (Example 2). This example uses substitution groups to have the ability to choose between two products: credit default swap or interest rate swap.

Example 2:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="FpML" type="FpML"/>
  <xs:element name="product" type="Product"></xs:element>
  <xs:element name="creditDefaultSwap" type="CreditDefaultSwap"
substitutionGroup="product"></xs:element>
  <xs:element name="interestRateSwap" type="InterestRateSwap"
substitutionGroup="product"></xs:element>
  <xs:complexType name="FpML">
    <xs:sequence>
      <xs:element ref="product"
maxOccurs="unbounded"></xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="Product">
    <xs:sequence>
      <xs:element name="name"
type="xs:string"></xs:element>
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="CreditDefaultSwap">
    <xs:complexContent>
      <xs:extension base="Product">
        <xs:sequence>
          <xs:element
name="referenceEntity" type="xs:string"></xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
  <xs:complexType name="InterestRateSwap">
    <xs:complexContent>
      <xs:extension base="Product">
        <xs:sequence>
          <xs:element name="interestRate"
type="xs:decimal"></xs:element>
        </xs:sequence>
      </xs:extension>
    </xs:complexContent>
  </xs:complexType>
</xs:schema>
```

InfoPath provides the ability to choose between the creditDefaultSwap or the InterestRateSwap. The tool is interpreting the behavior of the Schema correctly. (see Image 2):



We would expect to have a similar behavior using type substitution where InfoPath gives you the option of choosing the content model. This is not the case since InfoPath is assigning a type arbitrarily.

FpML uses type substitution not only in the root element but across the entire Schema. This is why solving this behavior would be useful to future implementations using FpML and InfoPath together.