

Is An "Other" Catchall Good Or Bad?

Issue

Should XML documents contain an "other" catchall element?

Background

Many XML instances (and their corresponding XML Schemas) may be found with an "other" catchall element.

Example

Suppose we are modelling the countries of the world. We might express the data model as an XML Schema:

```
<xs:element name="Country">
  <xs:complexType>
    <xs:choice>
      <xs:element name="Afghanistan" type="countryType" />
      <xs:element name="Albania" type="countryType" />
      ...
      <xs:element name="Zimbabwe" type="countryType" />
      <xs:element name="Other" type="countryType" />
    </xs:choice>
  </xs:complexType>
</xs:element>
```

Notice the last element declaration – it declares an "Other" element.

Since the countries of the world are ever-changing, we include an "Other" element in anticipation of some future, unforeseen country.

It is easy to imagine how a program might process an XML instance document that contains an actual country:

```
<Country>
  <Iceland>...</Iceland>
</Country>
```

It is less easy to imagine what meaningful actions a program might take when it encounters an "Other" country:

```
<Country>
  <Other>...</Other>
```

</Country>

This calls to question the usefulness of a catchall "other" element. The following sections argue both sides of this issue.

Reasons To Use An <Other> Element

A.1 Human limitations: there typically isn't enough time or money to perform an exhaustive investigation to create a list of all possible categories of data. Most attempts to categorize things in the real world are incomplete.

Example: if you attempt to produce a complete list of car makes and models, there will be someone who drives a car that is not on that list.

Example: If you ask someone their country of birth and offer a drop-down list, you'll almost certainly omit that country somewhere in the Balkans that existed for 3 months in 1926.

A.2. Some lists cannot be enumerated; sometimes the world changes; sometimes categories change; some things are one way in one part of the world but a different way elsewhere; some things requested should not be disclosed.

A.3 If you don't offer an "other" option, you are asking people to tell lies.

Example: Faced with being forced to enter a "country of birth" and a pick-list that is incomplete and does not contain the person's actual country of birth, people will lie and select an arbitrary country.

A.4 "Other" can be used as a feedback mechanism for the system to dynamically grow its knowledge: every time an XML document is received that contains an <Other> value, that value is added to the list of known values.

Reasons Not To Use An <Other> Element

B.1 It represents unexpected, unanticipated data.

B.2 It is an admission that you have failed to identify all the possible types of data.

B.3 It may be an indication that your business process is ill defined or your analysis is incomplete. An XML Schema should model the business process. When someone wants an "other" element or attribute value, it's almost always because the business process is ill defined, sometimes inadvertently, sometimes deliberately.

B.4 It may be an indication that your level of abstraction is incorrect. Stated another way, you may have picked the wrong level of granularity.

Example: if an XML Schema needs to change every time a country appears or disappears, that is going involve a lot of work keeping up with the real world. Instead of modeling the real world by actual country name:

```
<Afghanistan>...</Afghanistan>  
<Albania>...</Albania>  
...  
<Zimbabwe>...</Zimbabwe>
```

It may be better to have a <Country> element with a @name attribute:

```
<Country name="...">
```

B.5 There is little, if anything, that a machine or program can meaningfully do with unexpected, unanticipated data (i.e., with "other" data). That is contrary to one's objectives, which is to enable machine-processing.

Acknowledgements

Thanks to the following people for their contributions to this paper:

- Piotr Banski
- Manos Batsis
- Kurt Cagle
- Stephen Cameron
- Toby Considine
- Roger Costello
- John Cowan
- Bruce Cox
- James Garriss
- Norman Gray
- Jasen Jacobsen
- Rick Jelliffe
- Alexander Johannesen
- Michael Kay
- Bill Kearney
- Ihe Onwuka
- Thomas Passin
- Frank Richards
- Paul Silvey
- Michael Sokolov
- Oscar (Tei) Vives
- Lauren Wood