

## Highly Declarative Designs

Roger L. Costello  
February 2011

**Issue:** What are the characteristics of a highly declarative design?

I propose this as a characteristic:

If you can ask a question and get the answer by exclusively navigating markup, without any processing of data, then you have a highly declarative design.

**Example:** Consider the following XML Schema simpleType for constraining English language family name values:

```
<simpleType name="English-language-family-name">
  <restriction base="string">
    <minLength value="1" />
    <maxLength value="100" />
    <pattern value="[a-zA-Z' \.-]+" />
  </restriction>
</simpleType>
```

You can ask the XML Schema this question:

What is the longest English language family name allowed?

By exclusively navigating markup, using this XPath expression:

```
xs:simpleType/xs:restriction/xs:maxLength/@value
```

The question is answered:

100

Similarly, you can ask the question:

What is the shortest English language family name allowed?

By exclusively navigating markup, using this XPath expression:

```
xs:simpleType/xs:restriction/xs:minLength/@value
```

The question is answered:

1

Thus, with respect to questions concerning length constraints the above simpleType is highly declarative.

Contrast with the following simpleType definition, where the length constraints are part of the regular expression string:

```
<simpleType name="English-language-family-name">  
  <restriction base="string">  
    <pattern value="[a-zA-Z' \.-]{1,100}" />  
  </restriction>  
</simpleType>
```

If you ask the question:

What is the longest English language family name allowed?

You cannot answer the question by exclusively navigating markup. Processing of the regular expression string is required. For this particular regular expression, the processing that is required is not too great. However, answering the question for any arbitrary regular expression will require considerable processing. That's expensive and error-prone.

Ditto for this question:

What is the shortest English language family name allowed?

The latter simpleType is not a highly declarative design.

**Recommendation:** Wherever possible, use highly declarative designs.

For the example discussed above, the recommendation amounts to this: use the minLength and maxLength facets to express length constraint information; do not use a regular expression to express length constraint information.