

## The references in XML Schema are equivalent to GOTOs ... and GOTOs are evil

Roger L. Costello  
November 2012

**Requirements of XML Schemas:** An XML Schema needs to be readable, understandable and modifiable. One should be able to create a program that inputs an XML Schema and does processing on it or culls information from it.

**Assertion:** Using references in XML Schemas make it very difficult to achieve the requirements.

**Example Reference:** Consider this element declaration:

```
<xs:element name="Book" type="BookType" />
```

That type="BookType" looks innocuous enough, but let's recognize it for what it truly is: it is a reference, a pointer. It is pointing to a simple or complex type that is somewhere. But where?

Where could the referenced simple or complex type be located?

- It could be located in the same schema document as the element declaration
- It could be in an included schema document (child schema document)
- It could be in a schema document that is included by the schema document included by this schema document (grandchild schema document)
- It could be in the schema document that includes the schema document containing the element declaration (parent schema document)
- It could be a schema document that is included by the schema document that includes this schema document (cousin schema document)
- And the list goes on indefinitely (yikes!)

People developing XML Schemas do all of those.

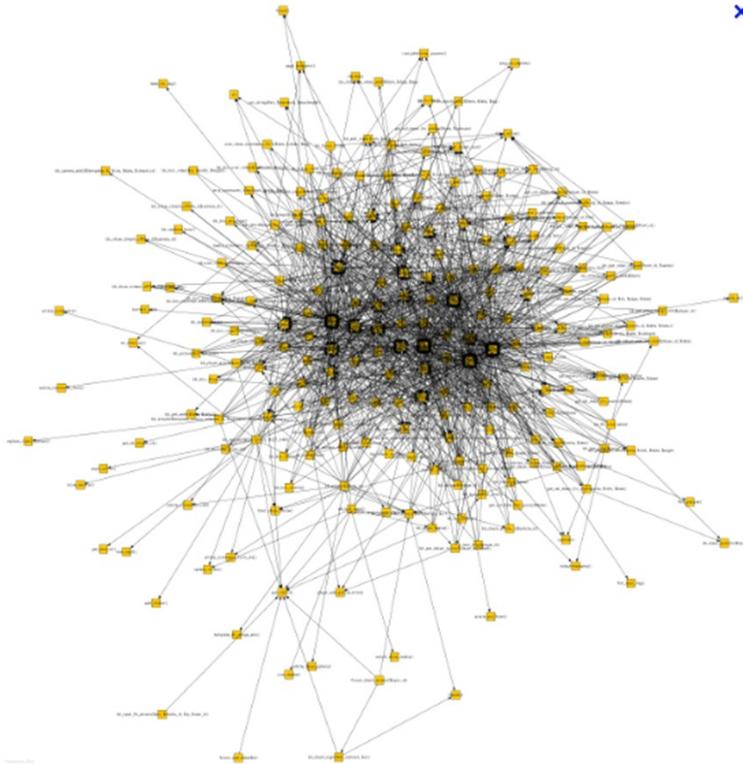
**Many Kinds of References:** The type="..." mechanism is just one mechanism for creating references/pointers. There are many others:

- xs:element ref="..."
- xs:attribute ref="..."
- xs:group ref="..."
- xs:attributeGroup ref="..."
- substitutionGroup="..."
- xs:restriction base="..."
- xs:import schemaLocation="..."
- xs:include schemaLocation="..."
- xs:redefine schemaLocation="..."
- xs:selector xpath="..."
- xs:field xpath="..."

There are probably others that I can't think of at the moment.

In a typical XML Schema there are hundreds or thousands of references/pointers flying around.

Here's a diagram that depicts the references/pointers in a typical large XML Schema:



Do you honestly think there is any hope of ever truly understanding, updating, or maintaining that tangled mess?

**Assertion:** The references/pointers in XML Schema are equivalent to GOTOs.

And all of us know that GOTOs are evil.

Therefore the references/pointers in XML Schema are evil.

So minimize their use. Ideally, don't use them at all.

**Recommendation:** create Russian Doll schemas.

“But then my schemas will be terribly large because chunks of schema code are repeated over and over.”

My response to that is this: If that is the case, then you need to rethink whether your schema is truly focused. Likely, it is a hodgepodge of things. Create small, tightly focused Russian Doll schemas. And use pipeline processing (XProc).