Recursive Markup

What is Recursive Markup?

Recursive markup is an element with a descendant element that has the same definition. Here's an example of a recursive definition:

```
A Section consists of a Title, Body, and optional Section.
```

And here's a sample XML instance:

```
<Section>
    <Title>The Question</Title>
    <Body>
        I sat perched on a small ledge, ...
    </Body>
    <Section>
        <Title>The Mysteries of Wealth</Title>
        <Body>
            What is wealth? ...
        </Body>
    </Section>
</Section>
```

The `Section` element is an example of recursive markup.

How to Define Recursive Markup using XML Schema

Recursive markup can be defined in two ways: (1) a `complexType A` contains a descendant element that has type `A`, or (2) an `element B` contains a descendant element that references `B`. Here's an example of each:

(1) The `Section` element is of type `SectionType`, which consists of the elements `Title`, `Body`, and an optional `Section`:

```
<xs:element name="Section" type="SectionType" />

<xs:complexType name="SectionType">
    <xs:sequence>
        <xs:element name="Title" type="xs:string" />
        <xs:element name="Body" type="xs:string" />
        <xs:element name="Section" type="SectionType" minOccurs="0"/>
    </xs:sequence>
</xs:complexType>
```

(2) The `Section` element consists of `Title`, `Body`, and an optional element that references back to `Section`:

```
<xs:element name="Section">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="Title" type="xs:string" />
            <xs:element name="Body" type="xs:string" />
            <xs:element ref="Section" minOccurs="0"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
```

When defining a recursive element, be sure that the element which recurses is optional; otherwise there will be infinite recursion. In the examples above, notice `minOccurs="0"`.

Recursively-Defined Information is Common

Recursively-described information is natural in many domains. Design XML around the inherent relationships of the information. Thus, if the information is naturally defined recursively, then implement it recursively.

A Few Examples of XML Vocabularies with Recursive Markup

XHTML: The `div` element is recursive. A `div` element can contain nearly any element, including `div`. Here's an example:

```
<div>
    <p>This is a description of CSS</p>
    <div>
        <p>Let's start with CSS selectors</p>
    </div>
</div>
```

Docbook: The Docbook vocabulary has several recursively defined elements. Here's one:

```
An orderedlist consists of listitem elements and each
listitem element consists of an orderedlist.
```

UBL: The Universal Business Language (UBL) has several recursively defined elements. Here's one:

```
The PriceList element consists of a PreviousPriceList
element, which has the same type as PriceList.
```

Support for Recursive Markup

Defining and processing XML documents with recursive markup is well supported by the XML technologies, including XML Schema, XSLT, XPath, and XQuery.

Some data binding tools may not support processing XML documents with recursive markup. If a tool cannot accommodate the inherent recursive relationships of the information, then switching to a tool that does support recursive markup is advised.

Acknowledgements

The following people contributed to this paper:

- David Allen
- Manos Batsis
- Roger Costello
- Craig Garrett
- Ken Holman
- David Lee
- Cecil New
- Liam Quin
- Michael Sokolov
- Andrew Welch
- James Winston