Published on XML.com Oct. 16, 2000
http://www.xml.com/pub/2000/10/16/relax/quickref.html

RELAX Quick Reference

Your entire specification must be wrapped in a `<module>`

```
<module
   moduleVersion="1.2"
   relaxCoreVersion="1.0"
   targetNamespace=""
   xmlns="http://www.xml.gr.jp/xmlns/relaxCore">

   <interface>
      <export label="root-element"/>
   </interface>

<!-- specifications follow -->

</module>
```

In the tables below, items in gray like this are optional. Items in italics *like this* indicate values that depend upon your particular application.

Specifying Elements

To specify:	RELAX

An empty element
 	
Note: all element rules may have a `label`. If omitted, it has the same value as the `role`
	`<elementRule role="name" label="name2">`
	`  <empty/>`
	`</elementRule>`

Element with no sub-elements	`<elementRule role="name" type="datatype"/>`

Element with constraints on content	`<elementRule role="name" type="datatype">`
	`  <constraint value="constraint-value"/>`
	`</elementRule>`
	Example:
	`<elementRule role="age" type="integer">`
	`  <minInclusive value="18"/>`
	`  <maxInclusive value="65"/>`
	`</elementRule>`

Element with a single sub-element	`<elementRule role="name">`
	`  <ref label="sub-element/>`
	`</elementRule>`

Element with sub-elements that must occur in a particular order
The `occurs=` attribute can have one of the following values:
`*` occurs zero or more times
`+` occurs one or more times
`?` occurs zero or one times
	`<elementRule role="name">`
	`  <sequence occurs="n">`
	`  <ref label="sub-element1" occurs="n"/>`
	`  <ref label="sub-element2" occurs="n"/>`
	`  </sequence>`
	`</elementRule>`

Element with sub-elements that may appear in any order	`<elementRule role="name">`
	`  <choice occurs="n">`
	`  <ref label="sub-element1" occurs="n"/>`
	`  <ref label="sub-element2" occurs="n"/>`
	`  </choice>`
	`</elementRule>`

Mixed content element (element can contain text that is not within any of the sub-elements)
The `<mixed>` element may enclose a `<sequence>` or `<choice>` as above.
	`<elementRule role="name">`
	`  <mixed>`
	`  <ref label="sub-element"/>`
	`  </mixed>`
	`</elementRule>`

A content model for re-use by other elements:
Hedge rules may refer to other hedge rules, but may not have mixed content or data types.
	`<hedgeRule label="hedge-rule-label">`
	`  <!-- content model as in elementRule -->`
	`</hedgeRule>`

Reference to a `<hedgeRule>`
You may intersperse `<ref>` and `<hedgeRef>` as required when defining an element's content.
	`<elementRule role="name">`
	`  <hedgeRef label="hedge-rule-label"/>`
	`</elementRule>`

Specifying Tags

Each element in a document is associated with tag. The `role=` in an `<elementRule>` matches the tag name.

To specify:	RELAX
A tag with no attributes All tags may have a `role`, which, if omitted, is given the same value as the `label`	`<tag label="name" role="tag-role"/>`

A tag with attributes
 	
You may specify that an attribute is required, or that it has a particular data type, or both.
	`<tag label="name">`
	`  <attribute name="attr-name-1"/>`
	`  <attribute name="attr-name-2" required="true"/>`
	`  <attribute name="attr-name-3" type="datatype"/>`
	`  <attribute name="attr-name-4"`
	`  required="true"`
	`  type="datatype"/>`
	`</tag>`

Constraint on an attribute	`<tag label="name">`
	`  <attribute name="attr-name" type="datatype">`
	`  <constraint value="constraint-value"/>`
	`  </attribute>`
	`</tag>`
	
	Example:
	
	`<tag label="water">`
	`  <attribute name="temp" type="decimal">`
	`  <minInclusive value="0"/>`
	`  <maxInclusive value="100"/>`
	`  </attribute>`
	`</tag>`

A set of attributes for re-use by other tags:	`<attPool role="pool-label">`
	`  <!-- list of attributes as in tag-->`
	`</attPool>`

Reference to an `<attPool>`
 	
You may intersperse `<ref>` and `<attribute>` as required when defining a tag's attributes.
	`<tag label="name">`
	`  <ref role="pool-label"/>`
	`</tag>`

Context Sensitivity

To make an element's content model dependent upon the element in which it is nested:

1. Define two `<elementRule>` models with the same `role` but different `label`s.
2. Other `<elementRule>`s may now contain the desired sub-element by referring to the appropriate `label`.

To make an element's content model dependent upon the value of its tag's attribute:

1. Define two `<elementRule>` models with the same `label` but different `role`s.
2. Create `<tag>`s with identical `name`s. Each of these will refer to the appropriate `role` defined in step one, and have an `<attribute>` to differentiate them.