

RDM-RELATIONS & XML-STRUCTURES AS XSETS

TABLES & RELATIONS AS XSETS

The following two RDM-Tables **T1** and **T2** have exactly the same 'domain names' and 'row values' and are therefore considered, under the tenants of the Relational data Model, to represent the same RDM-Relation. At an intelligent level of interpretation, this makes perfectly good sense. Capturing the same notions at a formal level is considerably more challenging.

	<i>A</i>	<i>B</i>	<i>C</i>		<i>C</i>	<i>B</i>	<i>A</i>												
T1 =	<table><tr><td><i>a</i></td><td><i>b</i></td><td><i>c</i></td></tr><tr><td><i>x</i></td><td><i>y</i></td><td><i>z</i></td></tr></table>	<i>a</i>	<i>b</i>	<i>c</i>	<i>x</i>	<i>y</i>	<i>z</i>			T2 =	<table><tr><td><i>z</i></td><td><i>y</i></td><td><i>x</i></td></tr><tr><td><i>c</i></td><td><i>b</i></td><td><i>a</i></td></tr></table>	<i>z</i>	<i>y</i>	<i>x</i>	<i>c</i>	<i>b</i>	<i>a</i>		
<i>a</i>	<i>b</i>	<i>c</i>																	
<i>x</i>	<i>y</i>	<i>z</i>																	
<i>z</i>	<i>y</i>	<i>x</i>																	
<i>c</i>	<i>b</i>	<i>a</i>																	

The leap of faith or the application of intelligence that renders the above two RDM-Tables as equivalent RDM-Relations requires that certain structural characteristics of the representation be ignored. Yet certain other structural properties may not be ignored. The challenge is to capture in a formal model just those properties (both structural and notational) that need to be preserved.

Notationally there is no challenge. All symbols used as references must already be unambiguous and uniquely well-defined. These exact same symbols must be preserved as is. Thus, in our example 'A,B,C,a,b,c,x,y,z' must also show up in a formal representation.

The structural challenge is in knowing what has to be preserved, what does not have to be preserved, and how to represent it so that the representation is self-contained, unambiguous, and with only one interpretation. That is, there is no need for a committee promulgation on how the representation should, could, or would be interpreted in order to please the committee.

First the answer in terms of 'scopes' and 'labeled elements', then an explanation of why it is an answer.

$$\begin{aligned}
 \mathbf{T1} &= \left\{ \{a^{<1,A>}, b^{<2,B>}, c^{<3,C>}\}^{<1>}, \{x^{<1,A>}, y^{<2,B>}, z^{<3,C>}\}^{<2>} \right\}. \\
 \mathbf{T2} &= \left\{ \{z^{<1,C>}, y^{<2,B>}, x^{<3,A>}\}^{<1>}, \{c^{<1,C>}, b^{<2,B>}, a^{<3,A>}\}^{<2>} \right\}. \\
 \mathbf{R} &= \left\{ \{a^A, b^B, c^C\}, \{x^A, y^B, z^C\} \right\}.
 \end{aligned}$$

XML-STRUCTURES AS XSETS

Given the following XML-Structures (ignoring the fact that they may be semantically vacuous in any known real world), can the structural relationship of the 'notational references' be preserved by a formal modeling as was done in the above RDM example?

```

<P>
  <p>
    <n>Alan</n>
    <a>42</a>
    <e>agb@abc.com</e>
  </p>
  <p>
    <n>Mary</n>
    <a>29</a>
    <e>mky@abc.com</e>
  </p>
</P>

```

Could be represented by:

$$\mathbf{P} = \left\{ \left\{ \{Alan^{<1>}\}^{<1,n>}, \{42^{<1>}\}^{<2,a>}, \{agb@abc.com^{<1>}\}^{<3,e>}\}^{<1,p>}, \right. \right. \\ \left. \left\{ \{Mary^{<1>}\}^{<1,n>}, \{29^{<1>}\}^{<2,a>}, \{mky@abc.com^{<1>}\}^{<3,e>}\}^{<2,p>} \right\} \right\}.$$

The use of 'scopes' and 'labeled elements' in the above example is intended only as introduction to the 'look and feel' of the notation. A formal explanation of the mathematical underpinnings, which will be offered later.

For a more bizarre example, consider the following:

```
<P>
  <p>
    <n>
      Alan
      <a>42</a>
    </n>
    <e>agb@abc.com</e>
  </p>
  <p>
    <n>
      Mary<a>29<e>mky@abc.com</e></a>
    </n>
  </p>
</P>
```

This example is not presented as one that is likely to occur in the reader's reality, but as one that is intended to stress the notational strength of the formal notation.

This can easily be represented using labeled elements, as follows:

$$\mathbf{P} = \left\{ \left\{ \{Alan^{<1>}, \{42^{<1>}\}^{<2,a>}\}^{<1,n>}, \{agb@abc.com^{<1>}\}^{<2,e>}\}^{<1,p>}, \right. \right. \\ \left. \left\{ \{Mary^{<1>}, \{29^{<1>}, \{mky@abc.com^{<1>}\}^{<2,e>}\}^{<2,a>}\}^{<1,n>}\}^{<2,p>} \right\} \right\}.$$

The Tags of XML-Structures may also contain attributes expressed as name/value pairs. This information can also be precisely captured as part of the label or scope associated with the appropriate element as in the following semantically stressed example.

```
<P>
  <p>
    <n a='42' e='agb@abc.com' >
      Alan
    </n>
  </p>
  <p>
    <n a='29' e='mky@abc.com' >
      Mary
    </n>
  </p>
</P>
```

The following demonstrates capturing 'attribute information' as part of an element's label or scope:

$$\mathbf{P} = \left\{ \left\{ \{Alan^{<1>}\}^{<1,n>}, \left\{ \{42^{<1>}\}^{<1,a>}, \{agb@abc.com^{<1>}\}^{<2,e>} \right\} \right\}^{<1,p>}, \right. \\ \left. \left\{ \{Mary^{<1>}\}^{<1,n>}, \left\{ \{29^{<1>}\}^{<1,a>}, \{mky@abc.com^{<1>}\}^{<2,e>} \right\} \right\}^{<2,p>} \right\}.$$

XML-Structures are not always as simplistic as these examples. However, more complex examples are just a matter of degree and do not depend on any new structural requirements. The following example includes some structural complexities that might naturally or theoretically arise in an XML-Structure, including embedded elements. Again the following example is intended to illustrate certain structural complexities and is not intended to be semantically justifiable.

```
<P>
  <p>
    <n>Alan<z>Z</z>G.<w>W<y d='H'>Q</y></w>Brown</n>
    <a>42</a>
    <e>agb@abc.com</e>
  </p>
  <p a="29" n="Mary" e="mky@abc.com" />
</P>
```

The following is one way to capture the mathematical identity of the above.

$$\mathbf{P} = \left\{ \left\{ \{Alan^{<1>}, \{Z^{<1>}\}^{<2,z>}, G.^{<3>}, \{W^{<1>}, \{Q^{<1>}\}^{<2,y>}, \{\{H^{<1>}\}^{<1,d>}\}^{>}\}^{<4,w>}, \right. \right. \\ \left. \left. \{42^{<1>}\}^{<2,a>}, \{agb@abc.com^{<1>}\}^{<3,e>}\}^{<1,p>}, \right. \right. \\ \left. \left. \emptyset^{<2,p>}, \{\{Mary^{<1>}\}^{<2,n>}, \{29^{<1>}\}^{<1,a>}, \{mky@abc.com^{<1>}\}^{<3,e>}\}^{>}\}^{>}\right\}.$$

These examples are not intended to be self-explanatory and will be treated in greater depth later in the paper. The examples are presented here as a preview of how it is possible to formally transform the 'vertical' structure-centric notation of XML-Structures into an informationally equivalent 'horizontal' operation-centric notation in terms of extended sets, Xsets.

Transforming a non-mathematical notation into a very formal arcane mathematical notation does not, in general, achieve much more than would be achieved by translating Mother Goose into Latin. It may look more impressive and convince some that it is now more academically sound, but the only immediate effect is that all the friendly intuition of the old notation has been destroyed.

Mathematical notation of and by itself does not guarantee mathematical soundness of the represented concepts. Any formal description of software development proves this point. For a hostile mathematical notation to be of sufficient practical value as to warrant the time and discomfort required make it friendly, it has to provide benefits far beyond its notational compactness.

The benefit in capturing the mathematical identity of XML-Structures in terms of Xsets is that all Xsets are indeed mathematically sound operands that behave very predictably under extended set operations, Xops. Since Xops are just extensions of set-theoretic operations that work on sets of 'labeled elements', all the mathematical power of set theory can now be harnessed to design, develop, implement, and use systems intended to process XML-Structures.