

Using the Concepts of Category Theory to Create Better XML Data Models

Roger L. Costello
August 2012

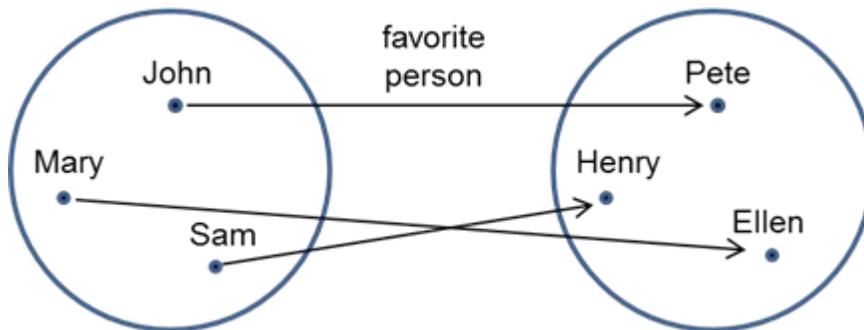
I am learning Category Theory [1] to enhance my data modeling skills and my functional programming skills.

Despite just starting to learn the subject, I have already discovered some exciting things that I will be able to use in XML data modeling and in XSLT functional programming.

Here's what I've learned:

Category Theory is about objects and mapping from one object to another.

For example, suppose you have two sets of persons and you want to map each person to their favorite person:

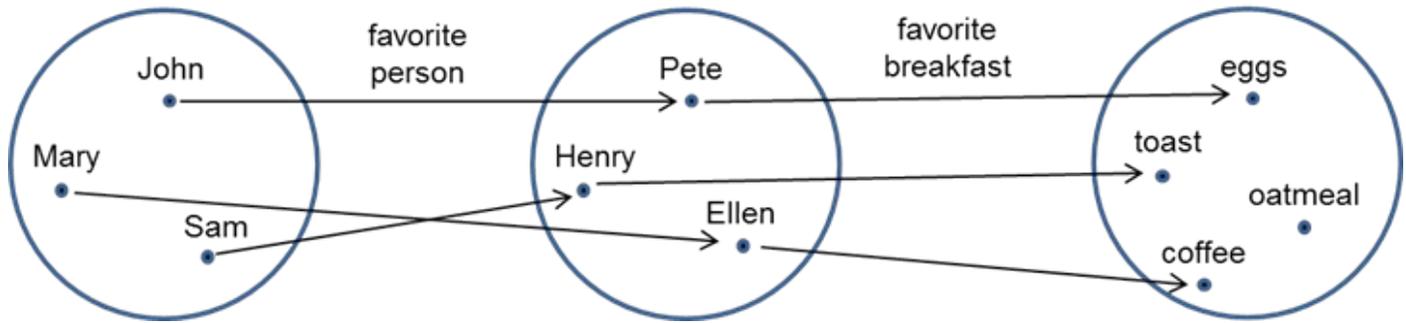


John's favorite person is Pete. Mary's favorite person is Ellen. And Sam's favorite person is Henry.

Fundamental to Category Theory is composition:

Category theory is based on composition as a fundamental operation in much the same way that classical set theory is based on the 'element of' or membership relation. [Barr, Wells]

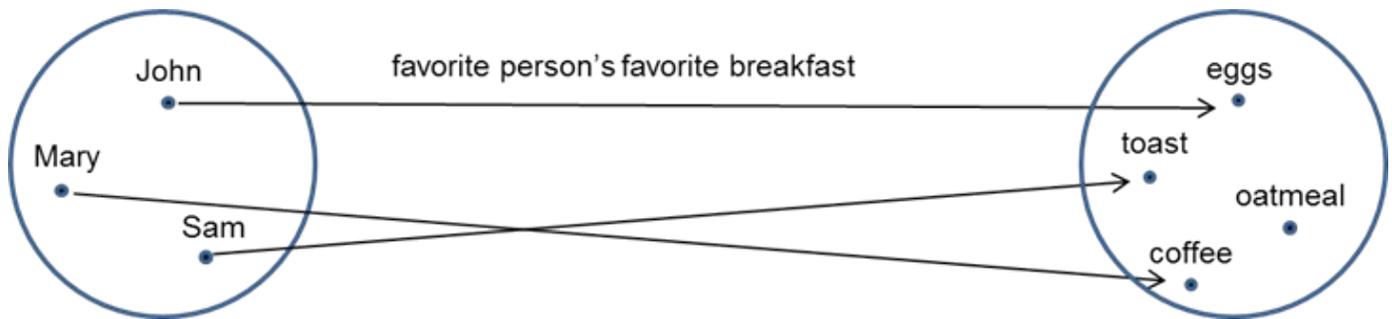
To illustrate composition, let's create a mapping from each favorite person to their favorite breakfast:



Pete's favorite breakfast is eggs. Henry's favorite breakfast is toast. And Ellen's favorite breakfast is coffee.

Now this is amazing:

We can compose the favorite person map and the favorite breakfast map to produce a new map:



The favorite breakfast of John's favorite person is eggs. The favorite breakfast of Mary's favorite person is coffee. And the favorite breakfast of Sam's favorite person is toast.

Neat!

Okay, so what this suggests in terms of data modeling is this:

Rather than modeling the data as a nested set of data:

```
Person: John
  Favorite Person: Pete
    Favorite Breakfast: eggs
Person: Mary
  Favorite Person: Ellen
    Favorite Breakfast: coffee
Person: Sam
  Favorite Person: Henry
    Favorite Breakfast: toast
```

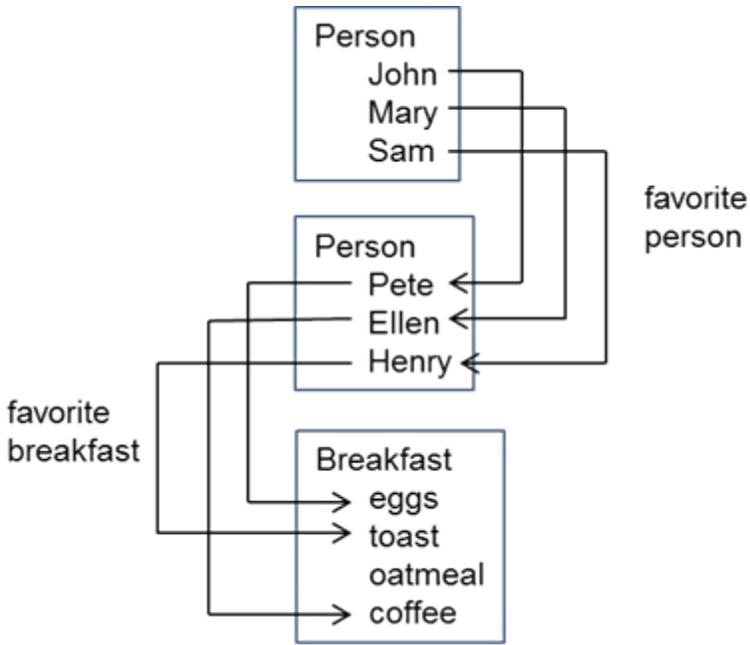
Instead model the data as independent objects:

```
Person
John
Mary
Sam
```

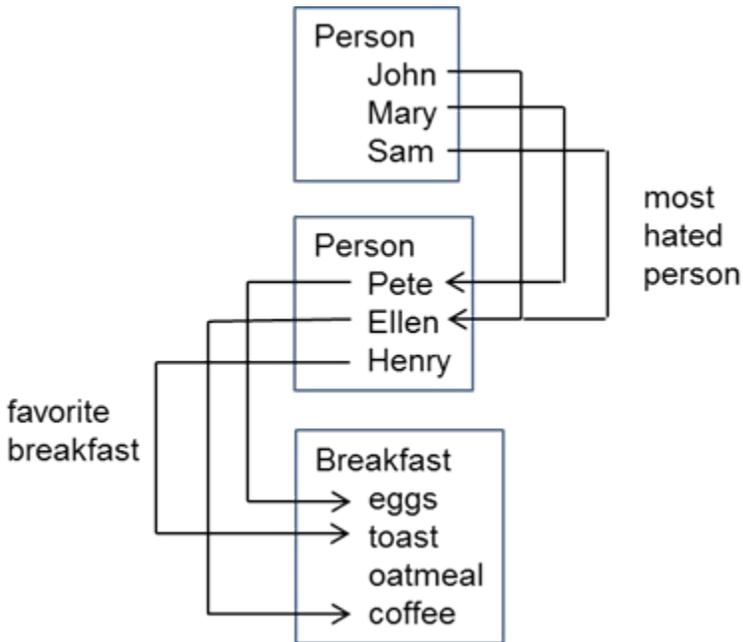
```
Person
Pete
Ellen
Henry
```

```
Breakfast
eggs
toast
oatmeal
coffee
```

And then apply the maps:



What makes this powerful is that we can apply different maps to the same objects. For example, let's replace the "favorite person" map with a "most hated person" map:



The favorite breakfast of John's most hated person is coffee. And so forth.

So, how would this be represented in XML?

Each element in an object has an ID attribute and the maps are expressed using an attribute of type IDREF:

```
<?xml version="1.0" encoding="UTF-8"?>
<Persons-and-Breakfast>
  <Persons>
    <Person id="john" favorite-person="pete">John</Person>
    <Person id="mary" favorite-person="ellen">Ellen</Person>
    <Person id="sam" favorite-person="henry">Sam</Person>
  </Persons>
  <Persons>
    <Person id="pete" favorite-breakfast="eggs">Pete</Person>
    <Person id="ellen" favorite-breakfast="coffee">Ellen</Person>
    <Person id="henry" favorite-breakfast="toast">Henry</Person>
  </Persons>
  <Breakfasts>
    <Breakfast id="eggs">Eggs</Breakfast>
    <Breakfast id="toast">Toast</Breakfast>
    <Breakfast id="oatmeal">Oatmeal</Breakfast>
    <Breakfast id="coffee">Coffee</Breakfast>
  </Breakfasts>
</Persons-and-Breakfast>
```

To change the maps simply change the IDREF attributes:

```
<?xml version="1.0" encoding="UTF-8"?>
<Persons-and-Breakfast>
  <Persons>
    <Person id="john" most-hated-person="ellen">John</Person>
    <Person id="mary" most-hated-person="pete">Ellen</Person>
    <Person id="sam" most-hated-person="ellen">Sam</Person>
  </Persons>
  <Persons>
    <Person id="pete" favorite-breakfast="eggs">Pete</Person>
    <Person id="ellen" favorite-breakfast="coffee">Ellen</Person>
    <Person id="henry" favorite-breakfast="toast">Henry</Person>
  </Persons>
  <Breakfasts>
    <Breakfast id="eggs">Eggs</Breakfast>
    <Breakfast id="toast">Toast</Breakfast>
    <Breakfast id="oatmeal">Oatmeal</Breakfast>
    <Breakfast id="coffee">Coffee</Breakfast>
  </Breakfasts>
</Persons-and-Breakfast>
```

This Category Theory stuff is fascinating.

[1] I am reading two books on Category Theory:

- a. This one is easy to understand: *Conceptual Mathematics* by Lawvere and Schanuel
- b. This one is harder to understand but more complete: *Category Theory for Computing Science* by Barr and Wells