DDI – SRG

Memo

To: The group raising the nested category proposal

From: DDI-SRG group

CC: **Date:** 03/22/2005

Re: Nested categories proposal

Summary

The Structural Reform Group has evaluated the proposal to reintroduce nested categories in the DDI 2.0. It is our view that the use of nested XML elements to represent the hierarchical structure of a dimension or a variable has several critical limitations. As an alternative, we have specified an alternative solution that as far as we can see meets all the requirements of the nested categories proposal with less radical changes to the existing DDI 2.0. This is also a solution which is more in line with the design principles of DDI 3.0 and as such might provide a smoother migration path.

The alternative proposal (an example)

The thrust of this solution is to use categories to describe leaf-nodes as well as parents (as in the nested categories model) but to use references (IDREFs) instead of nesting as a way to describe the hierarchical relationships. In most cases the use of IDREFs will provide a more flexible way of describing a hierarchical structure than XML nesting.

An example of how this might look using the example from the nested category proposal is listed below:

The first task is to describe the levels of the hierarchy (this is a slight tweak of the suggestion in the nested category proposal):

```
<catlevel ID=”Level1” levelnm="Broader sectors" />
<catlevel ID=”Level2” levelnm="Narrower sectors” />
<catlevel ID=”Level3” levelnm="Occupations" />
```

Note that we do not indicate nesting levels or roll-up structures here. This is done to be able to support ragged hierarchies. More semantics can of course be added to this element. This is just the minimal version.

As a second step all categories are listed in a flat structure (leaf-nodes as well as parents) . The hierarchical structure is represented by IDREFs referencing the children that belong to each parent. This is called downward referencing. As an alternative we could use upward referencing from child to parent. This would replace the proposed catgry attribute on the catgry element with a roll-up attribute referencing the parent(s) of a node. Note that there are no differences in functionality between the two methods. This is just related to design principles and efficiency (when it comes to parsing, etc.).

Each category will have a Level attribute that links back to the Level list above. Note that this way of doing it will support ragged hierarchies very nicely. It will also support level descriptions of leaf-nodes.

Note also that the structure will allow categories at the same semantic level to have different distance from the top-node. Leaf-nodes can also have different distance from the top.

```
 <catgry ID="C1" catgry="C2,C5,C6,C7,C10,C11 " Level=”Level1”>
   <catValu>0</catValu>
   <labl> Management, professional and related occupations </labl>
 </catgry>

 <catgry ID="C2" catgry="C3,C4" Level=”Level2”>
   <catValu>01</catValu>
   <labl> Management occupations</labl>
 </catgry>

 <catgry ID="C3" Level=”Level3”>
   <catValu>011</catValu>
   <labl> Top executives </labl>
 </catgry>

 <catgry ID="C4" Level=”Level3”>
   <catValu>012</catValu>
   <labl> Financial managers</labl>
 </catgry>

 <catgry ID="C5" Level=”Level2”>
   <catValu>02</catValu>
   <labl> Business and financial operations occupations</labl>
 </catgry>

 <catgry ID="C6" Level=”Level2”>
   <catValu>03</catValu>
   <labl> Computer and mathematical occupations</labl>
 </catgry>

 <catgry ID="C7" catgry=”C8,C9” Level=”Level2”>
   <catValu>04</catValu>
   <labl> Architecture and engineering occupations</labl>
 </catgry>

 <catgry ID="C8" Level=”Level3”>
   <catValu>041</catValu>
   <labl> Architects</labl>
 </catgry>

 <catgry ID="C9" Level=”Level3”>
   <catValu>042</catValu>
   <labl> Engineers </labl>
 </catgry>

 <catgry ID="C10" Level=”Level2”>
   <catValu>05</catValu>
```

```
  <labl> Legal occupations </labl>
 </catgry>

<catgry ID="C11" catgry="C12,C13" Level="Level2">
  <catValu>06</catValu>
  <labl>Education, training and library occupations</labl>
 </catgry>

<catgry ID="C12" Level="Level3">
  <catValu>061</catValu>
  <labl> Teachers </labl>
 </catgry>

 <catgry ID="C13" Level="Level2">
  <catValu>062</catValu>
  <labl>Librarians </labl>
 </catgry>
```

CatValu is present at all levels to allow efficient linking to data. For categories not represented by data, it will be possible to drop the catValu element. We also recommend a hasdata attribute to indicate whether a category is virtual or represented by data.

Again this is a minimum representation of the new model. Further elements should of course be included from the existing DDI.

Revised section of DDI 2.0 Summary Outline

The proposed changes will look like this:

```
|      |      |
|      |      |---- 4.3.18 catgry*     (ATT == ID, xml-lang, source, missing, missType, country,
|      |      |      |                         sdatrefs, excls, catgry, level)
|      |      |      |
|      |      |      |---- 4.3.18.1 catValu? (ATT == ID, xml-lang, source)
|      |      |      |---- 4.3.18.2 labl*    (ATT == ID, xml-lang, source, level, vendor, country, sdatrefs)
|      |      |      |---- 4.3.18.3 txt*     (ATT == ID, xml-lang, source, level, sdatrefs)
|      |      |      |---- 4.3.18.4 catStat* (ATT == ID, xml-lang, source, type, URI, sdatrefs, wgtd,
|      |      |      |                                wgt-var, weight)
|      |      |      +---- 4.3.18.5 mrow?      (ATT == ID, xml-lang, source)
|      |      |           |
|      |      |           +---- 4.3.18.5.1 mi*     (ATT == ID, xml-lang, source, varRef)

|      |      | - - 4.3.26 catlevel*     (ATT == ID, xml-lang, source, levelnm)
```

* new elements and attributes in red

Discussion related to DDI versioning and migration towards 3.0

The new model seems to be able to meet the majority of requirements, including the ones listed in the nested categories proposal.:

- It can easily support n-level hierarchies
- It will allow efficient description of levels all the way down to the leafs

- It has catValu elements connected to parents as well as leafs and can thus support efficient linking to data
- It will work fine for additive as well as non-additive cubes
- It will be able to represent ragged hierarchies of any complexity
- It will support multiple roll-ups (but would need an additional weight attribute to be complete)

"Nested categories" was raised as a validating change to the DDI version 2.0 and as a complement to the existing category group method. By accepting the proposal (or the above alternative) we are accepting that at least for a while (until 3.0) there is justification for two different methods doing more or less the same thing.

The goal for 3.0 is to come up with a model that meets all requirements (and thus eliminates the need for two models). Our proposal is therefore to introduce the method described above as a complement to category groups for version 2.0 (instead of the proposed nested categories). The new method can in the next round form the basis for the generic model of version 3.0 (nested categories can never do that). The proposed model can be added to 2.0 with less radical changes than nested categories.

By doing this we achieve three things:

- We meet the requirements of the original proposal
- We minimize the changes to Version 2.0
- We provide a migration path to 3.0 (where category groups hopefully can be taken out).

Recommended Usage

The SRG-group sees the alternative proposal as the way forward and the method that will form the basis for the description of hierarchical dimensions in 3.0. The new method should be used to describe any type of hierarchical dimension in 2.0. This will allow publishers and implementers to start preparing for the migration to 3.0. Note, however, that there may be situations in which using the category groups method makes the most sense to facilitate integration with legacy markup. The recommendation to use the new method will be added to the Tag Library.