

NESTED CATEGORIES PROPOSAL – COMMENTS LOG

Author of Comment	Reto Hadorn
Affiliation	SIDOS (Swiss Information and Data Archive Service For the Social Sciences)
Comment Text	<p>Dear colleagues,</p> <p>The proposal is obviously good, creating a flexible resource for various category structures. The comment below refers to an extension of the metadata structure, which may be necessary when it really comes to get support from software.</p> <p>What I am missing is a definition of the TYPES OF CATEGORY SYSTEMS which are expected to be represented with the tags proposed. Both the initial and the modified proposals ground the proposal on <i>examples</i>. They would be better grounded if they referred to <i>types</i>, since types could be defined by identifiable characteristics. Types can also be illustrated by examples; their advantage is that they unveil the dimensions on which category systems vary, and so lead us to some theory of category systems and give us a chance of covering more completely the 'space' of possible category systems to be represented in the metadata structure worked on.</p> <p>The types must be defined according to the functions and structural characteristics of category systems. This is the point where we must remember the life cycle stance: it is not just a matter of marking up textual codebooks, but also defining questionnaires, processing datasets etc. etc.</p> <p>We will meet a problem of similar nature while handling question structures; we will have to identify, define and name types of questions.</p> <p>There are two reasons for this.</p> <ul style="list-style-type: none"> - a flexible structure covering a broad set of needs does not give any control on the consistency of the metadata captured. If the type of structure is clearly identified and communicated to the software, the latter can support the user creating or capturing already defined metadata (the life cycle stance). - defining and naming structures gives the metadata creator, processor or user a better awareness of the nature of the objects he manipulates; knowledge starts with naming.

The question can still be asked, whether or not such information must be included in the metadata model; the reference to software creates the illusion that this is software-dependent. Actually, metadata structures should not be software-dependent. What we have to identify here are structures, the constraints of which are not fully defined within the proposed data model (the idrefs mechanism).

Let's start with naming the examples from the proposals:

The SINGLE LEVEL CATEGORY LIST

This is the classical case of response categories defined on one single level. No additional comment to be done.

The REGULAR CATEGORY GROUPING

SELF-EMPLOYED:

01. FARMERS, FISHERMEN (SKIPPERS)
02. PROFESSIONAL - LAWYERS, ACCOUNTANTS, ETC.
03. BUSINESS - SHOPOWNERS, CRAFTSMEN,

PROPRIETORS

EMPLOYED:

04. MANUAL WORKERS
05. WHITE COLLAR - OFFICE WORKERS
06. EXECUTIVES, TOP MANAGEMENT, DIRECTORS

NOT EMPLOYED:

07. RETIRED
08. HOUSEWIVES, NOT OTHERWISE EMPLOYED
09. STUDENTS, MILITARY SERVICE
10. UNEMPLOYED

A variable 'Municipality', grouped along counties and states is of the same type; it just has 2 grouping levels instead of 1.

The classification shows a hierarchy of logical levels, with data only on the lowest level. There may be more than two levels; critical is the fact that values are defined only on the last level.

A program can compute summing statistics for the groups at the various levels.

The IRREGULAR CATEGORY GROUPING

SELF-EMPLOYED:

01. FARMERS, FISHERMEN (SKIPPERS)
02. PROFESSIONAL - LAWYERS, ACCOUNTANTS, ETC.
03. BUSINESS - SHOPOWNERS, CRAFTSMEN,

PROPRIETORS

EMPLOYED:

UNDERDOGS

04. MANUAL WORKERS
05. WHITE COLLAR - OFFICE WORKERS

UPPERDOGS

06. EXECUTIVES, TOP MANAGEMENT, DIRECTORS

NOT EMPLOYED:

07. RETIRED
08. HOUSEWIVES, NOT OTHERWISE EMPLOYED
09. STUDENTS, MILITARY SERVICE
10. UNEMPLOYED

The number of levels is variable. Level 2 for example has sometimes the status of a 'section' sometimes of a category. This is the general case, of which the 'Regular category grouping' is a special case.

A program can compute summing statistics for the groups at the various levels. Yet, it must first analyse the structure to propose the right layout and make computations in the appropriate sequence.

RAGGED HIERARCHICAL CATEGORIES

- Management, professional and related occupations (Catgry C1)
 - Management occupations (Catgry C2)
 - Top executives (Catgry C3)
 - Financial managers (Catgry C4)
 - Business and financial operations occupations (Catgry C5)
- Computer and mathematical occupations (Catgry C6)
- Architecture and engineering occupations (Catgry C7)
 - Architects (Catgry C8)
 - Engineers (Catgry C9)
- Legal occupations (Catgry C10)
- Education, training and library occupations (Catgry C11)
 - Teachers (Catgry C12)
 - Librarians (Catgry C13)

	<p>I don't know how good such a classification is. I can occur when information comes at various levels of detail; you would probably put into C1 people which don't fit into any other category of this section.</p> <p>We have to ask what a program is able to do with such a classification, beside a fine layout in a publication of all categories. Probably it would be able to aggregate the lowest categories into the more inclusive category above, and this recursively, until a level of generality defined by the user is attained. This is not exactly the same operation as with the Regular and Irregular Category groupings, since the data for the 'aggregate' category must be added to the subcategories aggregated. This is why the structure must be declared to the software (and recored in the metadata)</p> <p>The challenge consists in proposing one set of elements capable of covering all requirements. The modified proposal may cover the requirements of all three types provided that</p> <ul style="list-style-type: none"> a) an element characterizes the type of category system for the variable at hand b) an attribute tells the program whether a category contains data or not. <p>So the existing catgryGrp could eventually be dropped. Actually, I don't think that the present catgryGrp element confers any power to a program to organise a category list for the regular category grouping; the list of categories grouped contains text instead of references, which does not allow for a fine layout together with statistics related to the categories and blocks the extension to multilingual solutions.</p> <p>The named 'Level' element is only appropriate in the case of the regular category grouping.</p> <p>Additional element: <catgryStr type =(singleLevel regular irregular ragged)/> Optional. singleLevel by default.</p> <p>Other types of category systems may have to be defined, and other uses also. The list of types must be extended and more systematically defined. The present comment is mainly a call for tackling the problem of declaring structures hidden in the very open data model of the DDI.</p>
Comment Date	04/04/2005
SRG's Response	<p><u>Dispensation of Comment regarding the inclusion of Additional element: <catgryStr type =(singleLevel regular irregular ragged)/> Optional. singleLevel by default :</u></p>

	<p>This is a bigger issue than can be addressed in a revision of the nested proposal. As stated in the comment "Other types of category systems may have to be defined, and other uses also. The list of types must be extended and more systematically defined." More work would have to be done here and this can be pursued in 3.0 or as a separate proposal for 2.0. Classification issues in general are a topic for 3.0.</p> <p>The Nested Category Proposal should proceed to the vote as it currently stands.</p>
Response Date	04/07/2005