Co-operative categorization in civil engineering

Marian Smutný, Jiří Kaiser
Czech Technical University in Prague – Faculty of Civil Engineering, Thákurova 7, 166 29 Praha 6, Czech Republic
marian.smutny@fsv.cvut.cz

Abstract. In the paper we indicate the way to allow the coexistence of multiple structures that are mutually supportive, but not mutually exclusive. It allows less organized users to classify entities according to their suitable structure, whilst leaving the possibility for specialised users to classify these entities into related field classification systems or domain ontologies, in mutually beneficial cooperation. Replacing the reductionist enumerative structure with multiplicative categorization should bring organizational and terminological flexibility that will allow future updates without disrupting the existing categorization. It allows one to use parallel hierarchical branches of categories according to aspects and classify the entity into several branches at the same time. It should allow users the freedom to create their own structures without disrupting the structures of other user groups. Such structures could coexist happily side by side. The aim is to indicate the model of an independent data organization system in building design in a multilingual environment, connected through diverse CAD, GIS, BIM, CAFM applications, ECM, CDE, emails, databases and file systems.

1. Introduction
In the paper we indicate the way to allow the coexistence of multiple structures that are mutually supportive, but not mutually exclusive. It allows less organized users to classify entities according to their suitable structure, whilst leaving the possibility for specialised users to classify these entities into related field classification systems (e.g. OmniClass, Uniclass, Cuneco, CoClass, etc.) or domain ontologies [1], in mutually beneficial cooperation. Commonly used principles of currently used classification systems are subject of various technical standards [2] (e.g. [3]). There are also various purposes of classification. E.g. to apply level of detail principles [4] or level of development principles [5], [6] for entities of BIM, we need to classify them. Also classification of objects in BIM libraries is necessary [7].

1.1. History
Since mankind began to accumulate written knowledge (information) in libraries, efforts had begun to find the necessary information, as quickly as possible. The recorded human knowledge had to be divided into smaller groups, i.e., categorized so that only those groups that are less numerous and relevant to the project can be searched. When there were too many of these groups, it was beneficial to group them, again according to their superior common attributes. In circa 400 BCE, Plato based a system of categories on a tree-like hierarchy. Fifty years later, Aristotle defined a category (class) and an entity (objects). In the years 1735-53, Linné defined an enumerative taxonomic categorization, which uses a reductionist hierarchy, i.e., an entity can be classified into only one category. It was not until circa 1930,
that the Indian librarian, Ranganathan, recognized the problems and limitations of categorization, based on taxonomic structure. He built his system on the idea that entities (documents) can be viewed from several aspects (facets), and created a faceted classification which uses a multiplicative hierarchy, i.e., an entity can be classified into multiple categories (Figure 1).

Multiplicative categorization allows one to use parallel hierarchical branches of categories according to aspects and classify the entity into several branches at the same time. The number of required categories will also decrease. If later needed to select all steel entities, simply mark this category in the material branch. The reorganization of the structure concerns only the categories in one branch and does not affect the structure of the others.

![Figure 1. Comparison of reductionist and multiplicative categorization](image)

**Faceted classification:**
1) The hierarchical structure has divided categories, from the most general concepts to the most specific ones [8].
2) The user does not need to a priori know the name of the category in which the object is placed. Categories expand with the number of entities with common terminology.
3) It allows one to assign more categories to an entity and allows them to be searched in more ways than a single predetermined order, e.g., screws can be searched for, either by material or by head type [9].

4) Aspects focus on important, essential, or enduring properties of the entities involved, which is especially important for categorizing rapidly changing content.

5) Multiple aspects can be combined in the search process, to quickly filter a set of entities. In addition, aspects can be used to address multiple categorization criteria [10].

6) New aspects can be created at any time without disrupting the original hierarchies or reorganizing other aspects.

7) Great flexibility means one needs to know little of the scope and organization of the domain in advance. It is difficult to break the faceted categorization scheme [11].

The most user-friendly way of batch categorization is to select entities with a common feature (aspect) and drag them to the appropriate category. Such a method of work (drag&drop) is close to human habits from the real world, for example when sorting a mixture of screws (entities) into different boxes (categories) according to various aspects - material, type of thread, shape of the head, etc.

![Figure 2. Example of categorization of graphic elements according to two aspects](image)

In Figure 2, at the top left, the elements represent engineering networks classified by dragging into the corresponding categories (corresponding colors) according to the aspect of specialization. Thus, the user selects elements with a common aspect (e.g., high-current engineering networks shown here with red elements) and drags them to the High-current category. In the other parts of Figure 2, the elements are classified once more, this time according to a schematic representation. These two aspects of categorization are intertwined.

After the elements are sorted, the user can filter the elements by tagging categories. Marking the Utility category (Figure 3 top left) will display all elements classified into the category and subcategories of this branch. Marking subcategories reduces the number of displayed elements. Figure 3 on the bottom right is an example of element filtering, which uses overlapping aspects and shows elements that belong to Electricity but are not Labels (use the logical operator AND NOT).
2. Assumptions

The solution we propose allows the coexistence of multiple structures that are mutually supportive but not mutually exclusive. It allows less organized users to classify entities according to their suitable structure, while leaving the possibility for specialized users to classify these entities, into related field categorization systems (e.g. OmniClass, Uniclass, Cuneco, CoClass, etc.) or domain ontologies, in mutually beneficial cooperation.

![Diagram showing filtering graphic elements in a hierarchy of categories.]

Figure 3. Example of filtering graphic elements in a hierarchy of categories.

2.1. Multiplicative categorization

1) Replacing the reductionist enumerative structure with multiplicative categorization (faceted classification ontology) should bring organizational and terminological flexibility that will allow future updates without disrupting the existing categorization.

2) It should allow users the freedom to create their own structures without disrupting the structures of other user groups. Such structures could coexist happily side by side.

3) An existing categorization of other users could be used advantageously as a pre-categorization of entities for a new categorization.

Multiplicative categorization allows one to use parallel hierarchical branches of categories according to aspects and classify the entity into several branches at the same time. The number of required categories will also decrease. If later needed to select all steel entities, just mark this category in the material branch. The reorganization of the structure concerns only the categories in one branch and does not affect the structure of the others.

To take this idea further, multiplicative categorization does not prevent the inclusion of the whole reductionist categorization, or the branch categorization system, as another parallel branch. This will allow one to combine different ways of organizing without being mutually exclusive.

2.2. External independent structure

1) Maintaining a structure in an external database outside the host application would allow one to work with uniform categories, regardless of the type of data, across the host applications. It would be possible to work with a uniform structure for both files and elements within applications.
2) The first condition is that host applications would provide an application interface for developing a component ensuring the host application's features are integrated with the categorization system.

3) The second condition is that host entities (files and elements) must support the retention of attributes that contain the written IDs of the categories to which they belong. If entities with attributes were converted, their categorization would be preserved.

Organizing entities is needed in every software application. The idea therefore arises as to whether the categorization structure could not exist uniformly and independently of software applications, and the type of their entities (graphic elements, documents, emails, computer files, etc.). Each professional area focuses on data exchange only in its own domain but fails to pass the organizational structure between domains.

Usually, the categorization structures and terminology are very similar, often overlapping with each other, at the level of document management (files, documents, emails, etc.) usually broader, and at the level of content management in professional parts (elements, paragraphs), usually finer. If users are accustomed to a certain structure of the data organization, then they are able to quickly find themselves even in an unknown application environment. The independent categorization system thus provides an organizational structure for any host application that uses them, with the help of the host component. Changes, updates and reorganizations of structures and terminology are thus reflected in all host applications.

The independence of the system is based on the assumption that entities will be able to retain the categorization attribute. Even if such entities are copied, transferred, or integrated into other files (even outside the categorization infrastructure), they will still carry information about their categorization. This assumption is based on the fact that meta-information is currently a common part of existing applications and file formats and is usually preserved in the manipulation of elements.

In other words, the categorization is carried by the entities themselves and the categorization persists throughout their existence, or until the categorization attributes are violated. The external structure gives the categorization meaning and serves for organisation and naming. Even if an entity is transformed as part of the conversion between different SW applications (e.g., CAD [12] elements are converted from MicroStation to Autocad), and the categorization attribute is also transferred, the transformed entity retains all its categorization.

![Figure 4. Two different categorization environments](image)

Entities (documents and elements) in the example on Figure 4, are classified into the categorization structure of ABECEDA (blue attributes). Some entities were copied or transferred to SW applications in
the company ALFABETA. Entities still contain ABECEDA category records in their attributes, but ALFABETA users cannot see them. ALFABETA users classify entities into their categorization structure (green attributes) and use their categorization.

If entities are transferred from the environment of one user, to the environment of another user, and their categorization structure is not found, the entities do not lose their categorization, even if their categorization is incomprehensible and unusable, without the categorization structure. If they return to their environment, it will again be possible to work with them, as classified entities.

Figure 5 shows an example where the ALFABETA structure was imported into the categories of ABECEDA. In the ABECEDA environment it is now possible to categorize entities into both structures, and also to filter entities according to the ALFABETA structure.

![Diagram showing categorization structures](image.png)

**Figure 5.** Inclusion of a categorization structure from a foreign categorization environment.

2.3. Multilingual names and semiotics

1) One and the same category could be displayed by different names.
2) File names in the file system should be replaced with an ID name and located in a single directory. Users should see the generated file names from their categorization. Static ID file names allow one to maintain links in applications, no matter what name is displayed to the user.
3) Name generation and synthesis would eliminate the current error rate, due to inattention or inconsistency that arises when manually naming files and directories.
4) In the form of categorization, it could also be possible to record any label into descriptive attributes for the elements.

One and the same category could be displayed by different names such as language variants, synonyms, abbreviations and codes. Entities categorized could inherit these names. Different user groups could thus see the same entities by their names, and in their organizational structure.

In the form of categorization, it could also be possible to record any labels in the descriptive attributes for the elements. The carrier of this value would be a category that contains a label in its semiotic attribute. The advantage of such notation would be the ability to represent one and the same value, to different labels. The current method allows one to search only the values used in the attributes, but not their synonyms (e.g., cover, cap, fastening, plug, lock, etc.) or alternative abbreviations or codes (WSV, UZV, EPO, etc.)

I suggest replacing the file names in the file system with a static ID name and placing them in one directory. Users should see the generated file names from their categorization, synthesized from
category names. Static ID filenames allow one to maintain links in applications, no matter what names are displayed to users.

Name generation and synthesis also eliminates the current error rate due to inattention or inconsistency that occurs when naming files and directories individually.

3. Results and discussions
Replacing the reductionist enumerative structure with multiplicative categorization should bring organizational and terminological flexibility that will allow future updates without disrupting the existing categorization. The assumptions of the indicated data organization system are multiplicative categorization and external application-independent structure. It allows one to use parallel hierarchical branches of categories according to aspects and classify the entity into several branches at the same time. It should allow users the freedom to create their own structures without disrupting the structures of other user groups. Such structures could coexist happily side by side.

The independence of the system is based on the assumption that entities will be able to retain the categorization attribute. That’s means the categorization is carried by the entities themselves and the categorization persists throughout their existence. Host entities (files and elements) must support the retention of attributes that contain the written IDs of the categories to which they belong. If entities with attributes were converted, their categorization would be preserved.

4. Conclusions
We suggest the way to allow the coexistence of multiple structures that are mutually supportive, but not mutually exclusive. It allows less organized users to classify entities according to their suitable structure, whilst leaving the possibility for specialised users to classify these entities into related field classification systems or domain ontologies, in mutually beneficial cooperation.

The aim is to design data's model of an independent data organization system in building design in a multilingual environment, connected through diverse CAD, GIS, BIM, CAFM applications, ECM, CDE, emails, databases and file systems.

Future research should find the way to implement the solution to various civil engineering software applications.

Acknowledgment(s)
This work was supported by the Grant Agency of the Czech Technical University in Prague, grant No. SGS20/103/OHK1/2T/11

References
[1] V. Nývlt, “Conceptual Meta Model for Building Information Modeling,” J. Phys.: Conf. Ser., vol. 1680, 2020.
[2] K. Prušková, “Beginning of Real Wide us of BIM Technology in Czech Republic,” IOP Conf. Ser.: Mater. Sci. Eng., vol. 471, 2020.
[3] ISO 12006-2:2015, “Building construction — Organization of information about construction works - Part 2: Framework for classification,” International Organization for Standardization, [Online] 2015, available from: https://www.iso.org/standard/61753.html.
[4] V. Nývlt, and K. Prušková, “Building Information Management as a Tool for Managing Knowledge throughout whole Building Life Cycle,” IOP Conf. Ser.: Mater. Sci. Eng., vol. 245, 2017.
[5] K. Prušková, “Reducing failures rate within the project documentation using Building Information Modelling, especially Level of Development,” MATEC Web of Conferences, vol.
146, 2018.

[6] K. Prušková, and V. Nývlt, “Issue of Building Information Modelling Implementation into the Czech Republic’s Legislation using the Level of Development,” IOP Conf. Ser.: Mater. Sci. Eng., vol. 245, 2017.

[7] M. Dědič, “3D scanning and analysis of acquired data of historically and culturally significant objects referring to the work of Adalbert Stifter,” MATEC Web of Conferences, vol. 279, 2019.

[8] J. M. Reitz, “Dictionary for library and information science,” Westport, CT: Libraries Unlimited, 2004.

[9] J. Sirovich, “Categories, Facets and Browsable Facets?,”, [Online] 2011, [Accessed 21-04-09] Available from: http://www.uxmatters.com/mt/archives/2011/08/categories-facetsand-browsable-facets.php .

[10] W. F. Godert, “Faceted classification in online retrieval,” International classification, vol. 18, pp. 98-109, 1991.

[11] P. Baofu, “The Future of Information Architecture,” str.104, 2008.

[12] G. Farin, J. Hoschek, and M. Kim, “Handbook of computer aided geometric design,” Elsevier, 2002.