The Process-Oriented Conceptual Model

Kaj A. Jørgensen

In the current revision discussion regarding ISO 12006-2, the following figure gives an overview of the primary concepts defined in the standard.

Figure 1 – Express G diagram with overview of basic concepts
This illustration is rather general but well understood by very many people in the construction industry and it covers many relevant concepts. The primary structure of the diagram is that construction processes have intake of resources and produces construction results of different types.

In the standard, it is also used beneficially as the basis for identification of areas for which it is relevant to develop classification tables and the result of this is also seen in a number of regional/national classification systems, e.g. OmniClass.

However, the question is: is this diagram the best foundation for discussion of development and use of classification tables in a model based approach, e.g. BIM, or should another kind of illustration be added? First of all, the diagram does not indicate very well that design, modelling and specification are about information/data. Hence, there is no clear view of the difference between real world (physical) objects and model data objects representing the physical objects. Primarily, it is the data objects that have to be handled. Secondly, the focus should be shifted towards the concepts, which represent the majority of occurrences in all construction phases (marked in Figure 2). In construction projects, one or a few construction complexes are identified and a fairly low number of construction entities are created. In contrast, a very large number of construction entity parts (components) must be handled. Of course, there are many occurrences of processes, i.e. tasks, activities, products and other resources but, until they are planned and carried out, data about these are related to the components data objects. In addition, a primary design/modelling focus must be set on spaces. Design and specification of these objects represent the largest work load so efficient and effective support here is mandatory.

![Figure 2 - The primary focus areas regarding design, modelling and specification](image)

In order to create a better illustration of primary issues of classification in a model based construction approach, the following sections underline the purpose of classification, how classification tables are used and how this is applied to modelling. The reasoning leads to Figure 9 as a possible illustration of major issues of classification related to a model based construction approach.

**Design and Specification – Modelling**

As indicated above, the use of classification tables is much related to design phase of construction entities and its components. As also argued, an efficient and effective use of classification tables is based on utilisation of appropriate software. This means that current

---

1 This concept has been discussed and will possibly be replaced by the concept 'element'. In order not to get into this discussion, the concept 'component' will be used in the following.
and future development of classification systems must take new trends in software
technologies into consideration. Today, the use of **modelling software** has gained
momentum and there is no doubt that this development will be continued and further
amplified. It must be underlined that the concept *model* is used in the general
understanding and not specifically computer based models. Any kind of description is in
principle a model.

The most promising type of modelling software is categorised as *object based*, which means
that the primary content of the models is a set of *software objects* representing the
commonly understood components of construction entities, e.g. walls, windows, lamps,
switches, sinks, pipes, and fittings. For these objects, it is often primarily the 3D geometric
shape that is represented in the software and, naturally, most modelling is about generating
such model objects and putting them together in order to visualise the overall architecture
of the construction entity. The general image of modelling is that it is rather constrained
regarding the support for detailing.

By use of modelling software, new model objects are selected based on in-built libraries of
object types. At first, such objects represent major sub-objects of construction entities but
subsequently the model components are detailed. Much development focus regarding
modelling software and model based software is about support for model detailing and this
means that an increasing number of tasks of the traditional specification process will be
integrated in the modelling process or in the use of dedicated software, which can utilise
models with various degree of detail.

The standard ISO/PAS 16739 can be considered as a foundation for creation of model
objects. This standard is about the buildingSMART data model Industry Foundation Classes
(IFC), which include a widely accepted taxonomy of object types, e.g. walls, columns,
distribution elements, sensors, products and tasks.

![Diagram](image)

*Figure 3 – Three dimensions of model object detailing*

According to general design theory, model components are detailed along three dimensions:
**sub-division**, **specification**, and **specialisation** (see Figure 3). **Sub-division** is to divide objects
into sub-objects, ultimately down to objects representing *products*. Initially, the internal
structure is given by the library types and the level of detail can be very different in the
software products. Likewise, the objects instantiated from library types are born with a pre-
defined set of attributes and some of them may have values assigned. Next, **specification** is
about adding further attributes to model objects and to determine values of attributes, e.g.
colour, material and fire resistance. Finally, **specialisation** is, from a given type, to select a
sub-type, e.g. from a generic door to select e.g. a sliding door. Unfortunately, specialisation
is not suitably supported by current modelling software.

Sub-division creates the **whole-part structure**, where the construction entity is subdivided
into components, which again are subdivided into other components etc. down to an
appropriate level. This is termed **composition** and the composition structure becomes a
**hierarchical structure** as illustrated in Figure 4.
As indicated, the composition structure is designer dependant and, hence, varies in different projects. Often, the composition structure is not very deep because current software does not support this very well. Generally, when components are sub-divided, the original object is deleted. IFC offers two kinds of decomposition relationships: aggregation of objects of different types and nesting of objects of the same type.

It is important to underline that, besides the composition structure, other structures can be created. Figure 5 illustrates how cross-going relationships between model objects can form the basis for other structures. In IFC, the relationship "contains in spatial structure" makes it possible to link space objects with construction entity objects and vice versa. Further, with assignment and connection relationships, a range of different cross-going links can be created. How much this is utilised in modelling software varies greatly.

Sub-division and specification is often performed through separate design of systems of the construction entity. Each system is primarily identified to provide a distinctive function and, accordingly, the system components often have specific mutual relationships. Each system is represented as a model object and the belonging components are established as a group as illustrated in Figure 6. This view of systems complies with IFC.
When a model has been created and it is used as the basis for production, the composition structure, other relationships and systems can be used to create an assembly structure and in connection with this structure, activities can be created. This is illustrated at the right side of Figure 7.

Classification tables can support modelling in three ways throughout the whole life period (see Figure 7): 1) when identifying and selecting types as the basis for new model objects, 2) when identifying sub-types, i.e. specialisation and 3) when performing specification modelling of the instantiated model objects.
As explained, model objects are created from object type libraries and the organisation of these libraries varies over software products. So, for users, it would be beneficial, if these libraries were based on existing classification tables of object types, ideally international standardised tables. By implementing a library of object types in software, the available types can be presented by generation of different structures based on the values of attributes. At least some of these structures should conform to standardised classification tables. In a typical design approach, the initial object type is rather general and, later on, this type may be changed to more special types by selecting a sub-type of the existing type in the previously used classification table.

Use of classification tables in specification modelling is about assigning codes, referring to entries of classification tables, to attributes of model objects. Current modelling software is not ideal for specification. Very often, this must be performed as a manual and time consuming task. Alternatively, by the use of separate software like analysis or simulation software, much specification can be generated implicitly. However, as previously stated, if tables in a classification system are related to each other, the specification operations can be much better supported. In order to use such an approach, it is an advantage that the library types have an attribute, which refer to a specific entry in the corresponding classification table.

This overall view of the use of classification tables is independent of what tools and technologies are applied to project but, in practice, users can be supported in many ways by use of dedicated software. Such software can support the use of classification tables in an automatic or semi-automatic mode, which is not possible in a pure document based approach.

Therefore, in order to make efficient and effective use of classification tables, it is important to use software, which can provide an integrated support throughout all phases. The basis for this is that relationships are established between tables in a classification system (illustrated in Figure 8). In practice, this means that codes, referring to one table, can support selection of codes in another table. This also implies that the first selected code is very important and, hence, should be one of the attributes/properties of the object, when it is created.

*Figure 8 – Relationships between classification tables can support efficient and effective work*

*Classification tables must enable support for modelling tasks by efficient and effective solutions in relationship with the use of modelling software. Key issues are about the degree of detail, which include further structuring, specification of models, and specialisation.*
**Model Object Types and Specification Datasets**

As stated, modelling includes selection of external data to be inserted into models. Figure 9 gives an overall range of datasets, which comply with general design and construction methodologies.

<table>
<thead>
<tr>
<th>Steps</th>
<th>Pre Programming</th>
<th>Programming</th>
<th>Integral Design</th>
<th>Detailed Design</th>
<th>Construction</th>
<th>Production</th>
<th>Operations &amp; Maintenance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spaces</td>
<td>User Activities</td>
<td>Functions</td>
<td>Safety</td>
<td>Other classifications and attributes/properties</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Components</td>
<td>Functions</td>
<td>Forms</td>
<td>Structures</td>
<td>Systems</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Products</td>
<td>Materials</td>
<td>Fire resistance</td>
<td>Environmental Impact</td>
<td>Technical Composites</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Activities</td>
<td>Other classifications and attributes/properties</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Project Specification</td>
<td>Functional Requirements</td>
<td>Space Program</td>
<td>Space Layout / Architecture</td>
<td>Structure</td>
<td>Estimates</td>
<td>Production Plans</td>
<td>O &amp; M Plans</td>
</tr>
</tbody>
</table>

*Figure 9 – Examples of datasets for specification of model object types*

Figure 9 shows a set of typical modelling steps (the top row) and important overall results of the work in these steps (the bottom row). The rest of the figure (the middle row) includes 1) primary model object types (middle row left column, including *spaces, components, products* and *construction activities*) and 2) typical examples of datasets (remaining part of middle row) to be used for specification of the created model objects. As stated, it is important to make this distinction because model object types are the basis ("templates") for creating/instantiation of model objects, the data objects/containers, while the datasets are the basis for specification, i.e. selected data from these sets are added to the model objects as attributes. This gives a clearer view of the individual datasets and makes the development of classification systems much easier.

It must be underlined that spaces, components, products and activities are by far not the only model object types but, to simplify, these are considered the most primary and other object types are subordinate and regarded mostly as implementation solutions. Some objects of those subordinate types are handled automatically by modelling software, e.g. objects for description of geometry, location and relationships.

As shown, the identified datasets are not necessarily related to specific modelling steps but are overlapping multiple steps and illustrate modelling progression according to general
design theory. In practice, many project approaches may be different regarding sequencing, degree of detail, resource allocation, etc.

Efficient and effective handling of these data is a key issue of modelling and, consequently, it is important to provide support of this. A beneficial support method is to develop classification tables for both each of the model object types and each of the specification datasets indicated in Figure 9. Such tables can then be used in the modelling steps as previously described. As already stated, IFC is an international standard with a widely accepted taxonomy of object types. As also stated and illustrated in Figure 8, it is important that classification tables are interrelated and also here, IFC can provide a foundation, i.e. mappings from the IFC object types to the tables for the specification datasets will support efficient and effective selection of specifications.

The key specification datasets for spaces are regarding user activities, functions, safety, accessibility, etc. Often, user activities and space functions are mixed but it is important to underline that they should be separated. Specification datasets for components, products and construction activities are many and some examples are shown in the figure. Forms and structures are important in the early phases and can support the specification of model objects with focus on the primary architecture and the overall organisation of the bearing system as well as the building services. Further, components may belong to different systems and may be specified by materials, fire resistance, environmental impact, etc. Technical composites primarily support the subdivision detailing process by specifying technical solutions down to the degree of detail, necessary for construction to be carried out. Some of the technical composites may be replaced by or linked to pre-fabricated products. In relationship with components, products and construction activities, datasets about trade/disciplines/skills, resources should be available for detailed specification. As also shown in Figure 9, other classification and attributes/properties may be added continuously over many steps.

**Classification Tables supporting Modelling**

Based on the description above and Figure 9, two categories of classification tables should be developed: 1) tables of primary model object types and 2) tables of concepts or datasets to be used for specification of created model objects.

**Primary Model Object Types**

As shown in Figure 9, there are four kinds of primary model object types: a) space types and aggregations, b) components and aggregations, c) activities/tasks and aggregations, d) products. For these object types various classification criteria (classification principles) can be applied.

Sample object types regarding spaces and aggregations of spaces are

List 1.a
- Space
- Zone
- Floor
- Section
- Level
- Storey
- Building
- ...

8
Sample object types regarding components and aggregations of components are

List 1.b
- Component
- Construction entity
- Construction complex
- Unit
- Plant
- ...

Sample object types regarding activities/tasks and aggregations of activities/tasks are

List 1.c
- activity/task
- operation
- job?
- work package
- project
- sub-project
- ...

In addition:

1.d: Classification of product object types

Concepts/Datasets for Specification of Created Objects

For these concepts or datasets, classification tables can be developed based on individually selected classification criteria/principles.

Sample datasets for specification of spaces are

List 2.a
- User activity
- Space function
- Safety
- ...

Sample datasets for specification of components are

List 2.b
- Function
- Form
- Structure
- System
- Material
- Fire resistance
- Environmental impact
- Technical composite
- Discipline/skill
- Resource
- ...
**Extraction and Structuring Data from Models**

An important subject about models is about structuring of data extracted form models. One directly available structure is the classification table, which have been used as the basis for creation of the model objects. However, as described, another backbone in descriptions or documentations of individual construction entities can be the *whole-part structure*, where the entity is subdivided into components, which again are subdivided into other components etc. down to an appropriate level. This structure is normally formed by the designer and, hence, becomes different in each project.

Composition structures of a model can be used for data extraction but, in different entity life phases, the most useful structure and the need for details may be different; so, a suitable structure as the basis for construction may not be ideal for e.g. the operations and maintenance. In this phase, for instance, detailed description of many basic components like the material layers of foundations, walls and slabs may be of minor importance while more detailed descriptions about coverings and entity service components may be of greater importance. Consequently, the most useful structure depends on the purpose.

Compositional structures can be generated on the basis of relationships between model objects (see Figure 5 and Figure 10) and it must be remembered that modelling software is supposed to create such relationships. If other kinds of software is used, e.g. traditional database software, relationships can easily be generated based on the geometry.

![Figure 10 – Simplified view of model object relationships according to IFC](image-url)

*General regulations* about, how descriptions and specifications of construction entities should be formed, will naturally also include provisions for the structure and the sequence in which description components should occur. The purpose of creating such regulations...
should be to standardise entity specifications and thereby to make it easier to share and compare entity specifications between partners.

Sometimes, such regulations are characterised as classifications but, seen from a theoretical point of view, they are not. Although they are of a general nature and they use names, which may be argued to represent classes, the overall characterisation is that they are about the whole-part structure of entities, where objects are grouped together and described/specified collectively. With reference to Figure 10, the two windows may be equal and grouped together and described once. Similarly, some description/specification of the four walls may be the same and, hence, refer to this group.

Because each model object has exact coordinates, the exact position in the entity on the site is directly available and can be extracted from the model. Such position data can be structured in multiple ways.