

Product Data Management and software automation in housing development: case Hartela Kide

I. Ville Rantanen

Department of Architecture
School of Arts, Design and Architecture
Aalto University
Espoo, Finland
ville.rantanen@aalto.fi

Abstract

This paper will explain how product data management and software automation were applied in a research and development project for construction company Hartela in Finland. The developed Kide-software is currently used by Hartela group to design and manage construction of dwellings, and as a tool for sales.

Building products used for construction and design of dwellings were divided into hierarchical categories based on their area of use, also defining the relations between products. A product database was established and integrated with the design software. Furniture layouts and pre-selected collections of products ("styles") were defined for common room categories and saved into the database. A verification algorithm was created to check that all product data definitions in the building information model (BIM) were correctly set and valid.

1 INTRODUCTION

The starting point for the work was a need to find an alternative practice for making substantial modifications to new buildings: it was usual, that kitchen cabinets were replaced and bathroom walls and floors retiled in newly completed apartments as the interior architecture did not meet the customers' expectations.

For customers, modifications caused additional cost and delay. For the construction company, extra work was needed and it was difficult to plan efficient schedules for consequent building projects as workforce were assigned to each project an unpredictable time period. To achieve a fast and efficient new construction process for customized apartments, a tool was needed for designing changes, calculating cost, automatizing purchases and the production of documentation and drawings.

More efficiency was also gained by standardizing product range and apartment designs. Software automation was created to assist design work as well as to create documents automatically. Allowing customers as much freedom as possible to make changes to the room layout within an apartment was also one of the key objectives. The pricing methods generally used in construction business had to be replaced with a new kind of pricing system: instead of compensating excluded products or work and calculating additional price for changes, products were priced with a fixed price. This required that general expenses of construction (design, marketing; construction of building exterior and common areas, dividing structures and structural members as well as HVAC systems) were included in a baseline price, whereas internal work and products were priced based on their direct expenses (purchase, transport and installation of products plus the profit margin of the company).

First buildings built with the new system were semi-detached houses and row houses. The interior of apartments was left to be built when a client had chosen an apartment and selected the room layout, surface materials and fixed furniture. An installation subfloor system was used to allow flexible positioning of sewers. Later, the system was applied to the construction of apartment buildings, too. In

this case, the whole building has to be finished on schedule. Thus, possibility to make changes to apartments is more limited as the construction proceeds. However, in Finland a general rule is that the majority of apartments have to be booked beforehand for a construction company to start actual construction of an apartment building. Therefore, most customers are able to take advantage of the possibility to customize their apartments with Kide-system.

2 BASIC PROPERTIES

As a basis for managing product data, a hierarchical model for building information was defined. Classifications were made for room types, product data items and grouping of product data items.

2.1 Room classification

An apartment was divided into 13 room types common in Finland. Combinations of product selections (styles) can be defined for each room type. In the database, a style definition contains tables for all room types.

Table 1. Room types

Code	Room
K1	Draught lobby
K2	Vestibule
K3	Lounge
K4	Kitchen
K5	Utility room
K6	Bathroom
K7	Toilet
K8	Shower
K9	Sauna
K10	Clothes room
K11	Living room
K12	Room in general (bedroom, workroom)
K13	Internal stairs

2.2 Global objects

General furniture and surfaces needed to model the interior of an apartment were specified as *global objects*. These objects are represented as parametric, geometrical objects within the CAD-software used (ArchiCAD). Global objects were designed as containers for product data.

Each global object is controlled with a script and groups of global objects are interactive. The spatial objects are a special group of global objects. They represent the surfaces of a room. Groups of other objects are dynamically interactive with spatial objects of the room they are positioned inside. For example, wall-related objects are automatically positioned according to the closest wall. Modifications to one object are inherited by other objects within the same room: for example, the height of toe-kick and cabinet worktop are related to each other to keep worktop upper surface level unchanged. This also affects installation height of wall cabinets and backsplash height to keep wall cabinets' and high cabinets' top surface at same level. Plumbing hook-ups, electrical switches and plugs are automatically positioned to correct height. They are also inserted together with appliances requiring electrical or plumbing connections.

2.3 Style definitions

The style definitions were created as database links connecting global objects' parameters to be defined with a link to a title in the database. A script is used to define relational conditions to determine which link should be used in case multiple entries could be linked with same parameter. For example, a primary link defining wall surface material class can be paint, tile, panel, glass or steel. A secondary link can be created for each material class. Depending on the primary selection, the correct secondary link is used. With this method, a preselected material can be defined for each material class when defining a style. In case the link is missing, a script is used to find a title from correct product category. Furthermore, each style defined item is verified and corrected if needed (See 2.9 Validation and verification). Style definitions can be saved directly from the BIM model with the integrated application, or edited with a browser-based interface.

Table 2. Global objects

Spatial objects	Accessories		Cabinets	Cabinet accessories
Floor surface	Window	Radiator	Base cabinet	Worktop
Wall surface	Door	Heat recovery unit	Wall cabinet	Sinks
Wall top surface	Threshold	Switchboard	High cabinet	Faucets
Ceiling surface	Lining	Sauna heater	Bathroom cabinet	Dishwasher valve
Fascia	Light, wall	Sauna benches	Mirror cabinet	Bidet hand shower
	Light, ceiling	Towel hook	Sliding door cabinet	End panel
	Switches	Toilet paper holder	Additional cabinet	Toe kick
	Wall plug	Wall surface area		Ventilation grill
	Floor drain	Shower cabinet		Backsplash
	Washer drain	Shower faucet		Filler
	Fireplace	Shower set		Under cabinet light
	Fireplace flue	Halogen light set		Hob
	Floor plate	Sauna light set		Toilet seat
	Ventilation vent, wall	Ventilation vent, ceiling		

The classification is an outcome of a pragmatic process of adding new classes for real products as needed. Therefore, the class hierarchy at present is still incomplete and not as logical and systematic as possible.

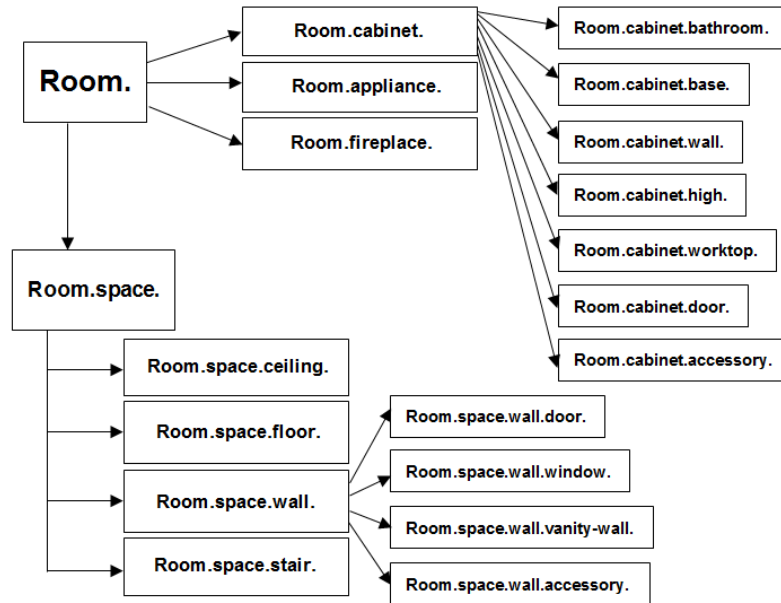


Figure 1. Classification of titles, main categories.

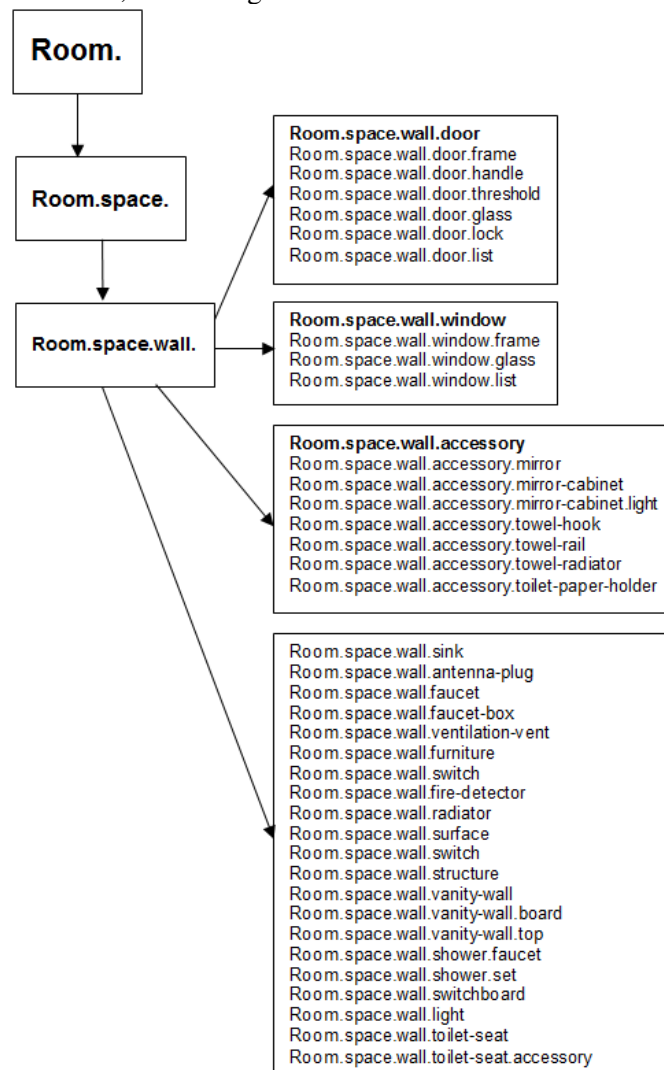


Figure 2. Classification of wall-related items.

Items related to walls are classified by their area of use. A category is defined for each product in database. Menu items in modeling software and style definition interface use category definitions to create selection lists. Global objects are parametrized with product data of corresponding category, including geometry and dimensional information for representation.

Based on the global object hierarchy, more detailed product data categories were created, defining an area of use for each title. Originally, each product title had a room category definition, too, but this was considered unnecessary and laborious to maintain.

2.4 Title definitions

Each item (title) in database contains a unique identification code, textual representation, classification category, pricing unit, purchase and selling price, supplier, period of validity, dimensions. Also colour, texture and 3-dimensional geometry for presentation are contained in the database, as well as any other information or documents like installation guides. Titles can also contain a relation and group definition. Relation includes one or more categories of child titles, whereas group definition is used to define groups of items from different categories.

2.5 Relation definitions

A typical relation definition connects master element with child elements. For example, a wall cabinet can have child elements like cabinet door, under-cabinet light, integrated cooker-hood and microwave oven, the category class of which can be defined with a relation definition.

2.6 Group definitions

In addition to the class hierarchy, we found out a need for a group definition. Instead of adding numerous new levels to class hierarchy, a group definition was used to define groups of elements in different categories. For example, a cabinet door type is usually available in certain colours only. The door type and colours available were given the same group definition, allowing an algorithm to be used to show valid choices and verify set values. Making the class hierarchy more complex would make user interface more difficult to use, and the database would have consisted of several duplicate items located in different categories. Group definitions can overlap, so different combinations of items in one category can be linked with several items in another category.

2.7 Leader titles

It turned out that major amount of titles were actually different kitchen cabinet doors, as each size of each door type has to be represented as a unique item. To make the list of kitchen furniture shorter and more useful in practice, each door type was defined with a leader title. A leader title is a general title belonging to a certain group of titles, and it has no size or price information. In our naming system, it was identified by adding ".leader" to the category name. When lists of elements in the BIM model are generated, cabinet doors are listed with corresponding leader title. Leader titles can be used to define styles, too, as the verification algorithm will transform leader title to correct unique titles based on dimensions and group definition of the element.

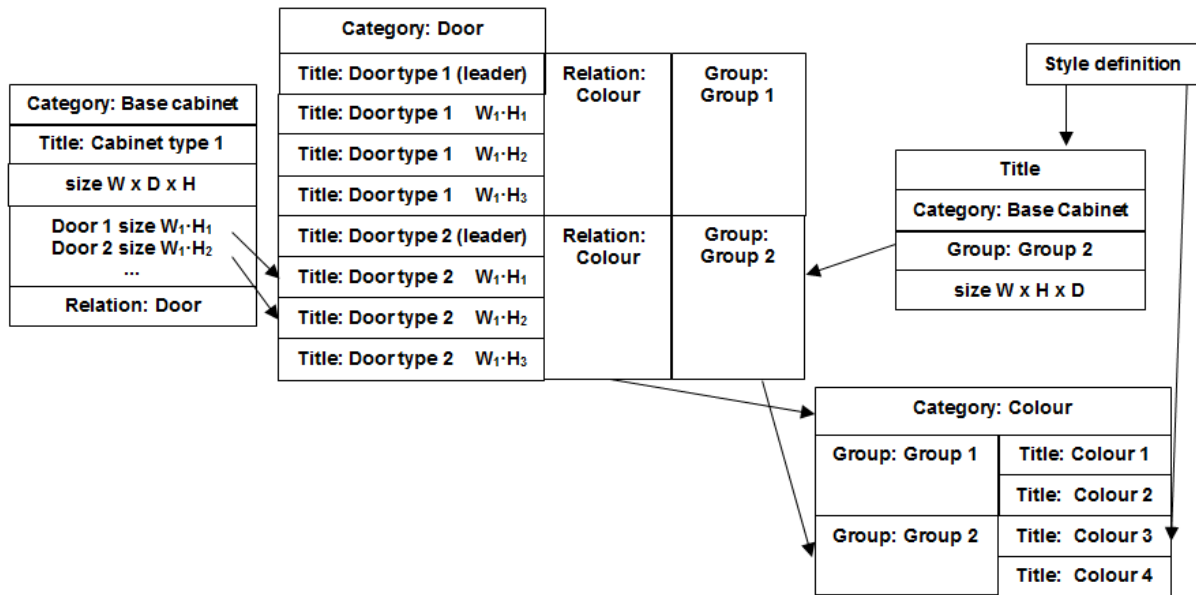


Figure 3. An example of relation and group definitions defining a valid title selection based on style definition.

The style definition sets a title with certain attributes. If the object parameters are contradictory, a best match is searched based on relation and group definitions.

2.8 Validation and verification

Users can modify the BIM model and data contained in it by making style selections, by changing single attributes with user interface menu selections, or by graphically editing the model. Any of these methods will change parameters of objects so that the title defined may be incompatible with other attributes of the element. A verification algorithm was created to check title definitions of elements.

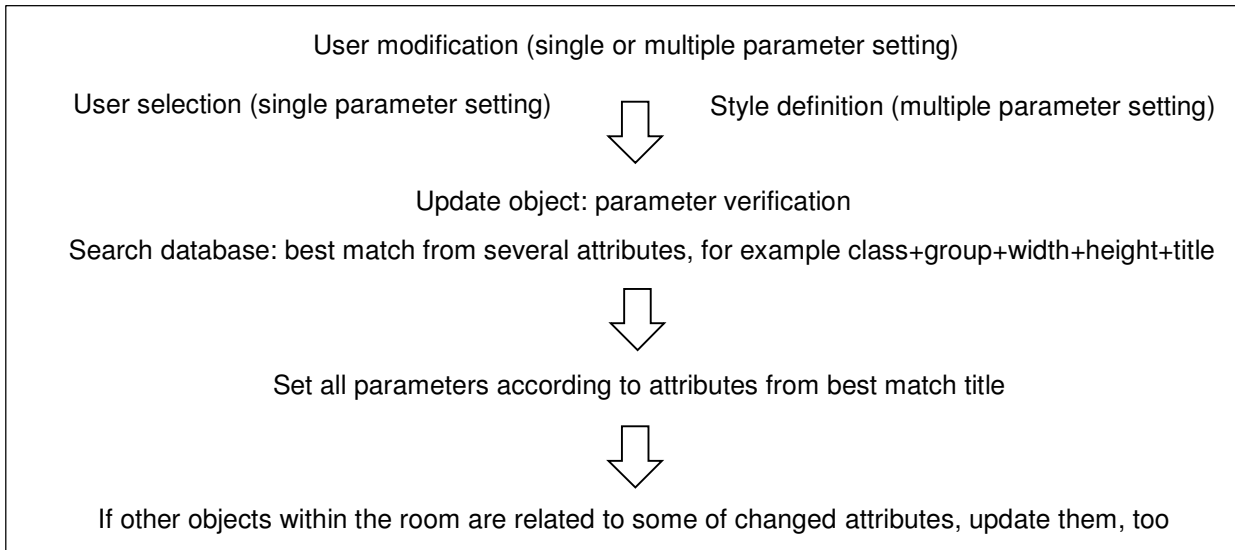


Figure 4. Verifying title definitions (product data embedded into the BIM model).

BIM model object parameters can be changed by style selection affecting all objects within a room; by making menu selections or by graphical editing of objects by the user. Modified parameters are verified from database with iterating rules.

2.9 Replace method

In the verification process, a flexible search for best matching replacement title was needed. In some cases, the groups of titles were inconsistent, and it was not possible to find a best match within a group. To cope with this situation, a replace method was introduced. When searching for the best match, alternative categories can be defined.

```
<search>category=[relation];height=[height];depth=[depth];group=[group:bestmatch:replace(Group1|Group2):replace(Group3|Group4):replace(Group5|Group6)]</search>
```

Figure 4. An example of the script used to search titles in the database using replace method.

Search algorithm will search for a title with exact category, height and depth definitions, but with a group definition looking for a best match with possibility to replace the group with another group.

3 SUMMARY

This paper provides a compact description of the dilemmas encountered and solutions developed in a research and development project that has been going on for the past 10 years. The main observation might be that the real world needs and requirements are quite complex and require several methods of data management to tackle with. There is a continuous seek for balance between the ease of use and maintenance, and the need to define more complex classifications, relations and interactions to emulate and represent reality.