

# GTC mapping rule specification

## 1 Introduction

The GTC hierarchy consists of fixed levels and rule based levels. The fixed levels are maintained and delivered by GTC governance organization, and in each class, there is a mapping rule/rule name specifying the criteria for selecting products belonging to that class. The rule base levels come after the fixed level. These levels are affected by the assortment each tool vendor has. GTC governance organization provides abstract classes and mapping rules in this level, and each tool vendor needs to implement the classes based on the mapping rules and their own product assortment.

	Class Name	Mapping Rule
Classes in fixed level	▲ Cutting tool library	select {ctl}
	▷ Tool item	select {tntp}
	▲ Adaptive item	select {ait}
	▷ Rotation symmetrical adaptor	select [RPMX]>0 AND NOT {collet}
Abstract classes in rule based level	▲ Collet	select {collet}
	▲ Machine side [connection interface]	define {connect_mach}
	Workpiece side [connection interface]	define {connect_wkps}

Figure 1. Example of GTC generic hierarchy provided by GTC governance organization

	Class Name
Fixed level	▲ Cutting Tools from Sandvik Coromant
	▷ Tool item
	▷ Insert
	▲ Adaptive item
	▷ Rotation symmetrical adaptor
Rule based level, abstract classes are implemented	▲ Collet
	▲ Collet CoroGrip/HydroGrip (SZC02)
	Cylindrical clamping (ZYL01)
	▲ Collet DIN6345 (Rubberflex) (SZD11)
	Tap shank clamping (ZYL21)
	▲ Collet DIN6388 (SZD01)
	Cylindrical clamping (ZYL01)
	Tap shank clamping (ZYL21)
	▲ Collet ER/DIN6499-B (SZD02)
	Cylindrical clamping (ZYL01)
	Tap shank clamping (ZYL21)
	▲ Cylindrical drill shank (ZYL11)
	Cylindrical clamping (ZYL01)
Cylindrical drill shank clamping (ZYL11)	
▲ Cylindrical shank (ZYL22)	
Cylindrical clamping (ZYL22)	
▲ Whistle Notch (ZYL13)	
Cylindrical clamping (ZYL01)	

Figure 2. Example of GTC vendor hierarchy

This document will explain how the mapping rule expression should be interpreted. By applying the mapping rules on product data, tool vendors generate their own GTC rule based level hierarchy and assortment file which tells where a product fits in the hierarchy. Pre-condition is that the product data shall base on ISO13399. A product needs to be classified to one or several of the PLib item type

classes (include ait, astp, cins, titp and their posterity) and cutting operation classes (include ctp and its posterity), and product parameters are described by PLib properties. .

## 2 Rule expression

The syntax is intended to be human readable and understandable. Tool vendors are encouraged to translate them into queries/logics which fit their platforms to develop automotive mapping engines.

For the expression of mapping rules, the following notational conventions apply:

- `{}` Represents an ISO13399 PLib (part-library, version released on 2008) class, enclosed text is a class short name. E.g. {titp}= item class “tool item type”.
- `[]` Represents an ISO13399 PLib (part-library, version released on 2008) property, enclosed text is a property symbol.  
E.g. [IIC]=property “insert interface code”, [RHO]{fdp}=property “rho” in the name scope of “feed direction primary”, [RHO]{YZWP}=property “rho” in the name scope of “yzw-plane”. Figure 6 shows an example of PLib properties and name scope class in product data file (.p21 format).

**select** A filter applied on a set of items which result from parent filters. E.g.

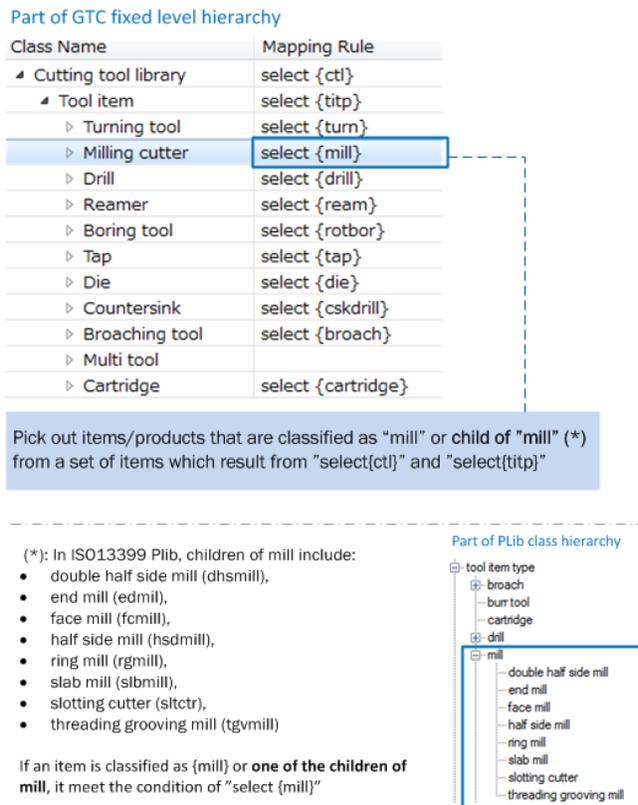


Figure 3. Example of ‘select’ notation

**define**

Used on abstract classes in rule based level, specifying the rule for generating GTC class nodes based on vendor's own assortment.

**Example 1:** define GTC classes based on connection interface

▾ Adaptive item	select {ait}
▸ Rotation symmetrical adaptor	select [RPMX]>0 AND NOT {collet}
▾ Collet	select {collet}
▾ Machine side [connection interface]	define {connect_mach}
Workpiece side [connection interface]	define {connect_wkps}

Figure 4. Example of rules based on connection interface

This type of rule based hierarchy is to group products by their connection interface features. First, take products which have the same **machine** side connection interface into one group, and then, divide the group into sub-groups based on the group members' **workpiece** side connection interfaces. The number of GTC classes generated in this level depends on the vendor's product variants. Figure 8 shows the detail process and naming convention for each class.

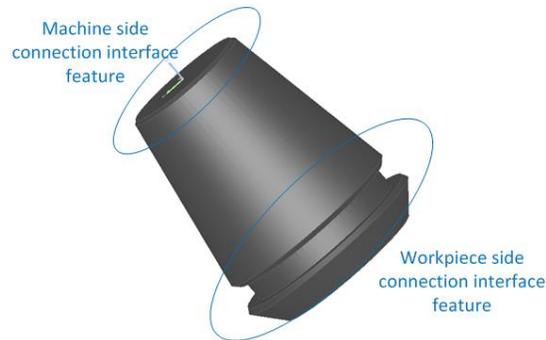


Figure 5. connection interface on machine side and workpiece side

In a product data file which uses STEP format (.p21) and EXPRESS Schema-Cutting\_tool\_schema\_arm (defined in ISO13399-1), the connection interface feature on each side is represented as in Figure 6. (When [SIDE]=mach, the name scope class is “variant 02 of SZD collet connection standard”, which means this is the connection interface on machine side. When [SIDE]=wkps, the name scope class is “variant 01 of ZYL cylindrical shank connection”, so this is the workpiece side connection interface.)

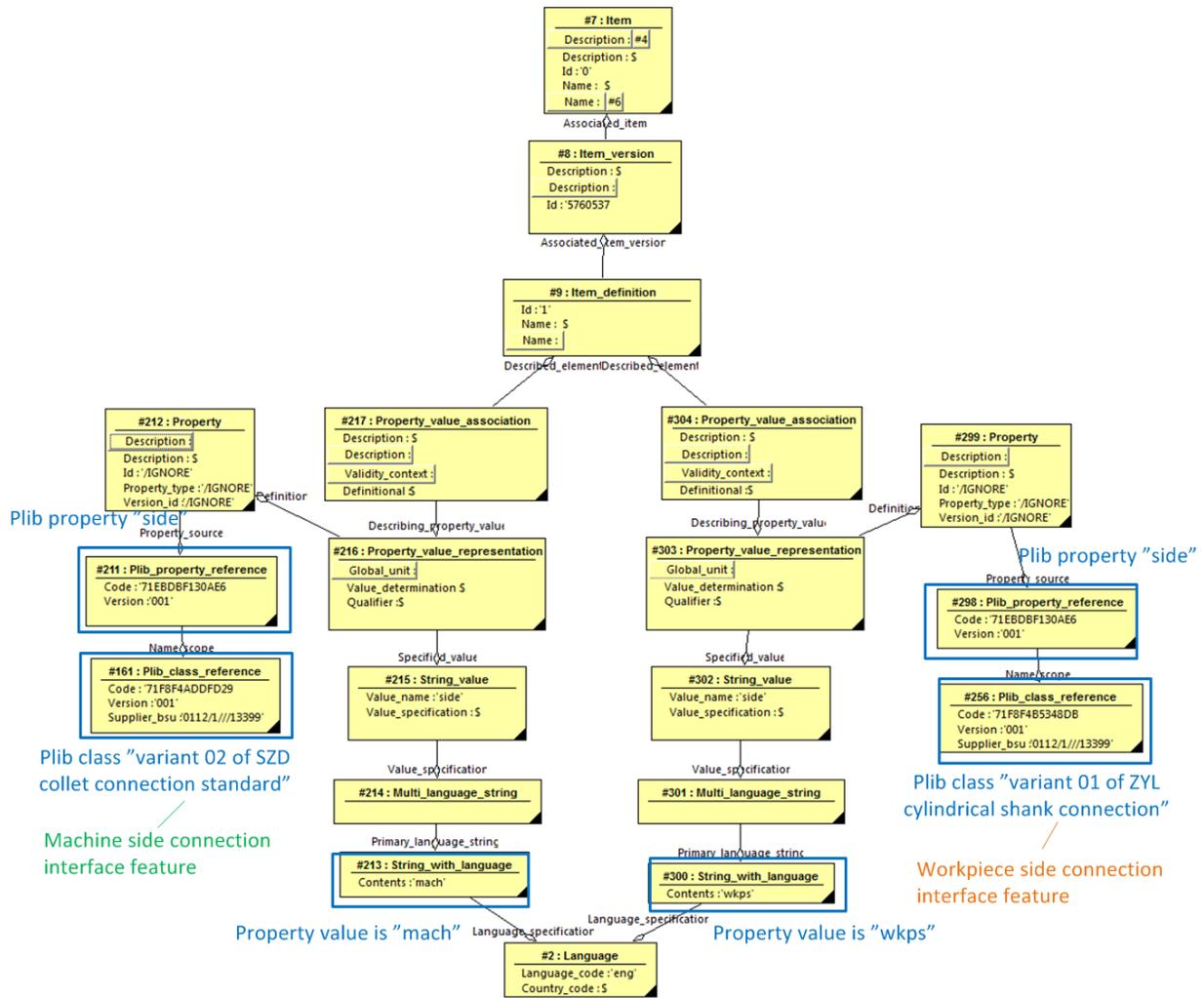


Figure 6. STEP data structure for PLib property [SIDE] and its name scopes and values.  
 ( The EXPRESS schema – cutting\_tool\_schema\_arm is defined in ISO 13399-1 Annex C)

	01	02	03	04	05	06	07	08	09	10	11	12
IIC	ISO	C	N	12	04	00	2	*	*	*	*	**

ISO insert concept

	01	02	03	04	05	06	07	08	09	10	11	12
IIC	SV	M	H	17	04	00	3	R	*	\$	\$	**

Non-ISO insert concept

Figure 7. IIC codes and symbol allocations (for more details, please check ISO13399-60:2014)

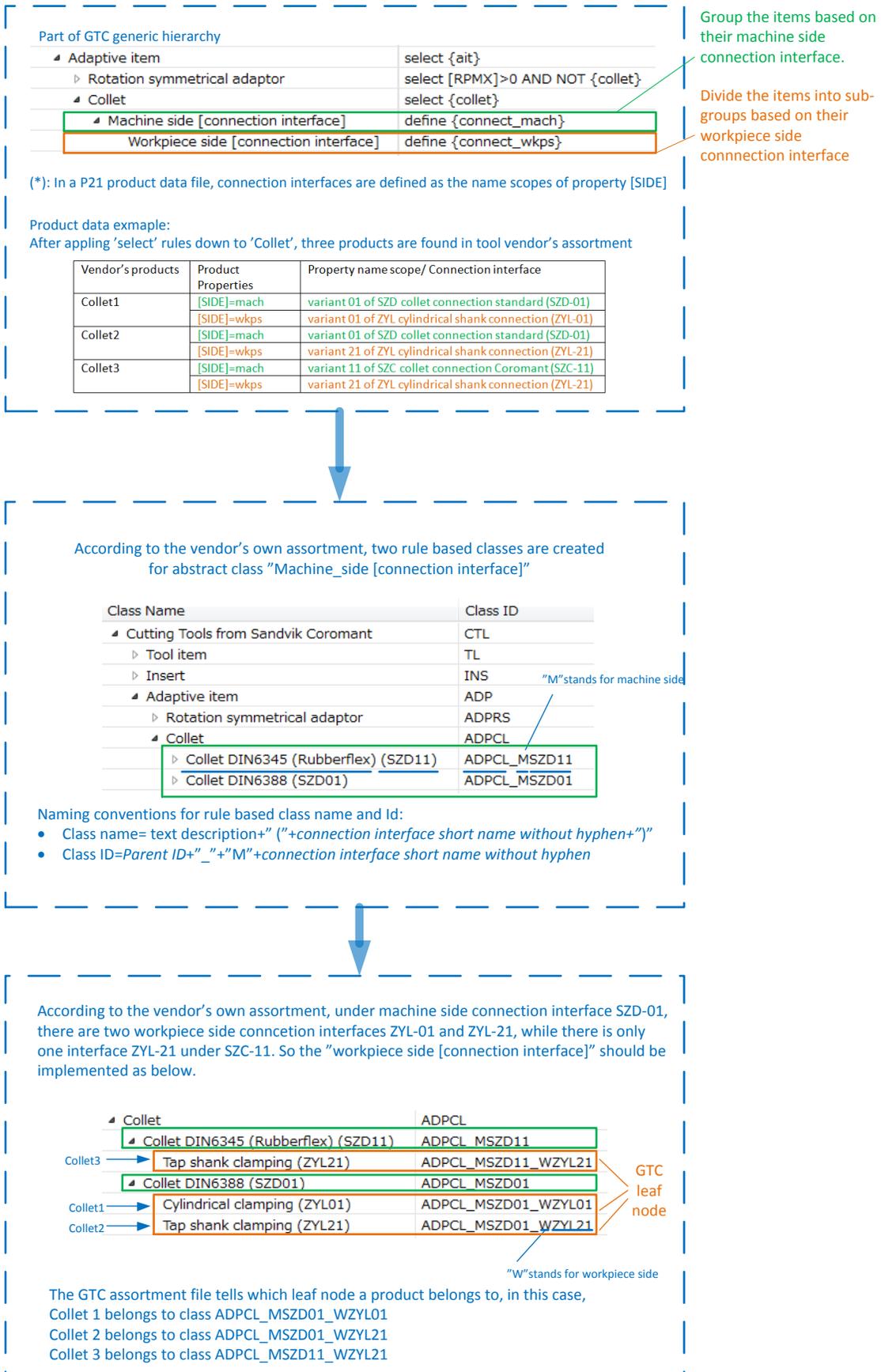


Figure 8. Detail process of defining classes based on connection interface.

**Example 2: define classes based on insert concept (\*)**

(\*) Details about insert concept can be found in ISO13399-60:2014, ISO 1832:2012, DIN 4000-100

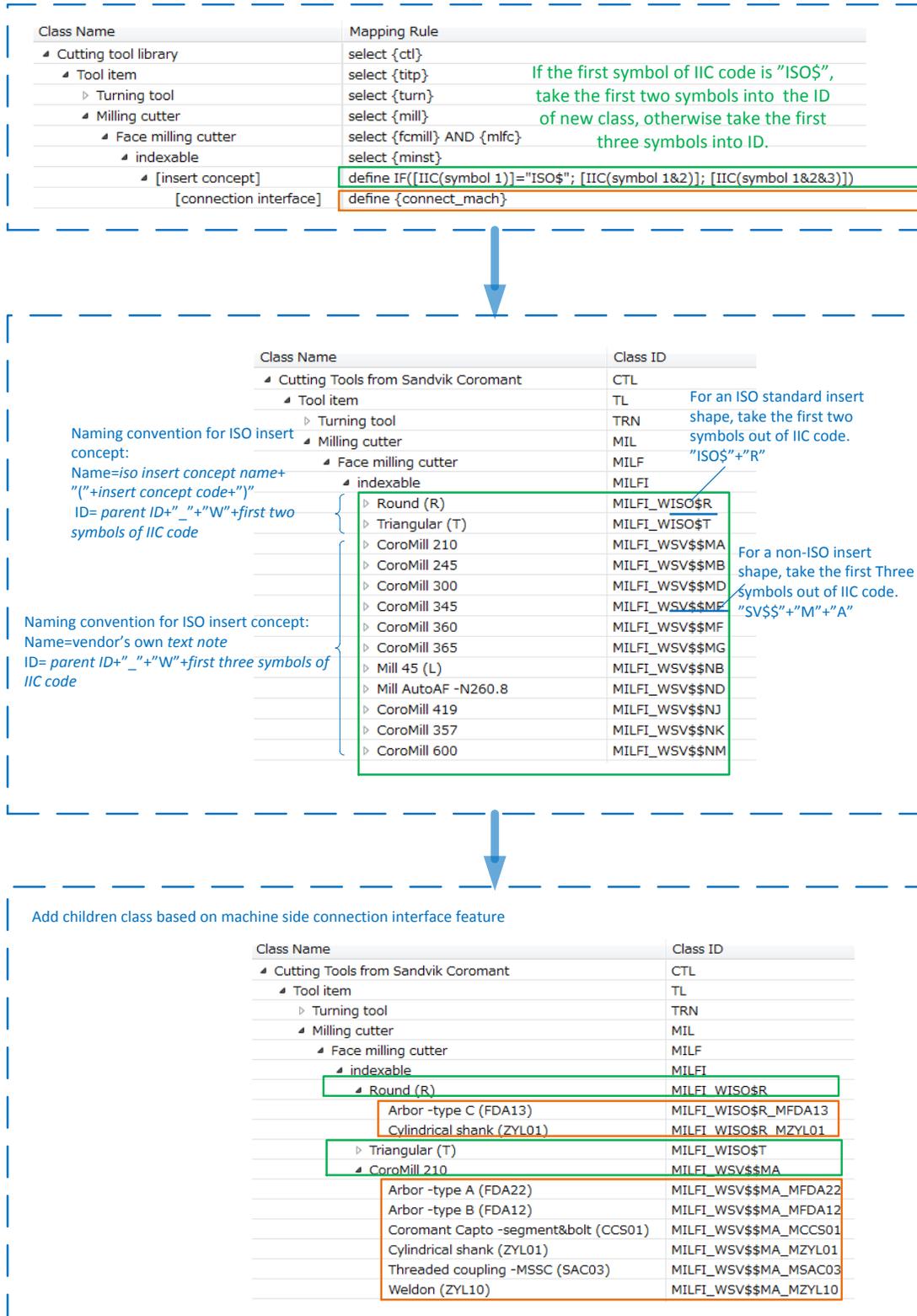


Figure 9. Example of rules based on insert concept.

For symbol allocations in IIC code, check Figure 7. For reading machine side connection interface out of product data in .p21 format, check Figure 6. For generating classes based on connection interface, check example 1.

### **Logic operation**

“AND”-logic AND, “OR”-logic OR, “NOT”-logic NOT

“=”-equal, “>”-larger than, “<”-less than, “<>”-not equal

**E.g.1** “select {thext} OR {thint} AND NOT ({gvg} OR {part}) “  
= select items which are classified as/are used in (\*) “threading external” or “threading internal” but not “grooving” or “parting”

(\*): A PLib class can be used as classification or name scope of a property (see Figure 10), the rule is valid also if a child of the PLib class is used by the item.

**E.g.2** “select {edmil} AND {mlprf} AND ROUND([DC]/[CEMR],1)=2.0”  
= select items which are classified as/are used in “end mill” and “milling profile” (both classes are used), and the result of “cutting diameter” divided by “cutting edge major radius” is within the range of [1.95, 2.05) (round up by 1 decimal is equal to 2.0).

**E.g.3** “NOT [GEP]” = property GEP is not used or its value is false. (GEP is a Boolean type)

**E.g.4** “NOT [CDI]{minst}” = property CDI is not used within the name scope of {minst}. (CDI is not a boolean type, so just check if it is used)

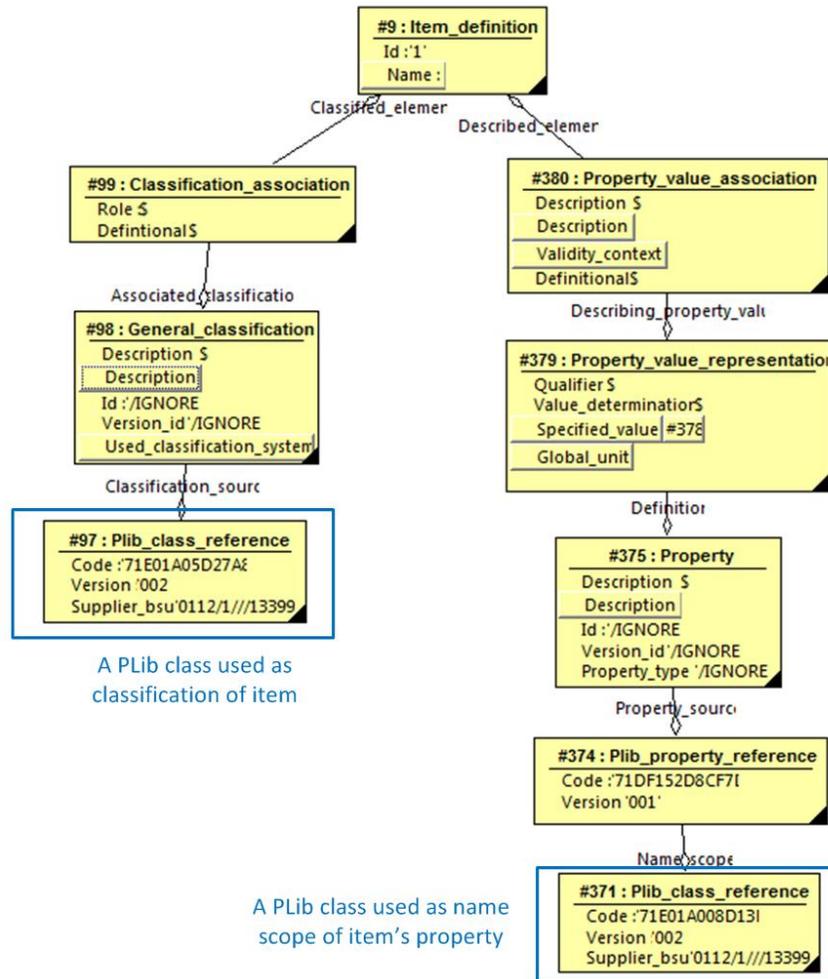


Figure 10. A Plib class can be used as classification or name scope of a property

**Calculation function** Functions used in mapping rules is MS excel functions

**Empty** A generic and netrual mapping rule is not able to be defined due to limitable of PLib version 2008. A tool vendor can specific its own rule when implemented.

**select true/select false** A generic and netrual mapping rule is not able to be defined due to limitable of PLib version 2008. Hardcode products to the class which has '**select true**'. A tool vendor can replace it with its own rule when implemented.