Introduction into GTC Format Implementation

Version 1.0

Date: 31.03.2015

Author: Moka Badertscher, Interstate University of Applied Sciences of Technology Buchs NTB

moka.badertscher@ntb.ch

Content

1. Abstract

The purpose of this document is to provide a general introduction into the implementation of the GTC (Generic Tool Catalog) format. This format is a vendor-neutral catalog structure which enables the exchange of cutting tool data between vendors and applications. It is developed mainly by product data and application experts. The implementation goal for vendors is to provide their cutting tool catalog as a GTC catalog package with defined content and structure. The GTC format comple-ments the ISO 13399 standard. The **product files**, which are basic files within the catalog structure, follow the ISO 13399 standard. They contain references to explicit referenced data which is available as a library dictionary. The dictionary provides explicitly defined classes and properties. The **assortment file** contains the list of products with the details concerning the exact classification in the GTC hierarchy. The implementation goal for the application side is to be able to read and interpret this catalog data according to the GTC format. The main part of the **GTC hierarchy** definition consists of a generic class id for each hierarchy level and of mapping rules, which are used for defining which class a product belongs to. In this document fundamental concepts of ISO 13399 standard are explained as well as different aspects of GTC format definition and individual steps for the GTC implementation on vendor and application side.

2. Introduction

The business value of having a common digital format for cutting tools is clear to both tool vendors who would like to create data only once and to applications that would like to read all cutting tool catalogs in the same format. Details can be found in the "Sandvik Coromant Technical White Paper: GTC Guidelines" [D_12], which describe the purpose and business value of GTC Generic Tool Catalog and offers well-founded introduction into the topic of GTC. This document offers a general introduction into the implementation road of GTC format, which consists of a vendor-neutral catalog structure and the GTC hierarchy. The GTC format is being developed as a complement to ISO 13399 standard and shall make the digital catalog exchange possible. This general introduction covers the GTC format definitions and ceratin ISO 13399 concepts as well as the interaction between GTC and ISO 13399.

GTC has been adopted by the major tool vendors like Sandvik, Kennametal and ISCAR over the last years. This adoption extends to the software vendors like Siemens PLM, the tool data management provider TDM and to CimSource, which provides digital tool services. The target audience of this document are newcomers to GTC which could benefit from experiences of the forerunners.

3. The existing standards and technologies

This chapter explains the standards and technologies which are used in the context of cutting tool data exchange. Chapter 3.1 shows what kind of data is exchanged and chapter 3.2 is an introduction into the main concepts of ISO 13399 standard with its representation model and parts library. The actual implementation however, can only be done through a detailed study of the ISO 13399 documents or support by experts.

3.1 Product data exchange

The product data in the cutting tool area is exchanged via STEP-files (.step, .stp, .p21) and described by the model language EXPRESS. EXPRESS is defined in ISO 10303-11 and the STEP-file format is defined in ISO 10303-21. In the remainder of this document product data files will be referred to as p21-files. The EXPRESS schema for the cutting tool area is defined in the schema ISO 13399-1, Annex C.

3.1.1 Item Types

A vendor catalog contains many p21-files and each one of them describes one item. As shown in Figure 2 an item has four concrete manifestations: cutting item, tool item, adaptive item and assembly item. Assembly items, however, are not yet considered in the GTC hierarchy. The different categories are defined as follows:

A) A cutting item is either an insert or the cutting edge of a solid tool, it holds information about the cutting portion of a tool.
B) A tool item holds a cutting item. It may be the tool body of an assembly or the body of a solid tool.
C) An adaptive item modifies the properties of an assembly and cannot hold a cutting item by itself, e.g. an adaptor or extender.
D) An assembly item is a component necessary for the complete assembly of a tool. It may change properties of components but does not change the properties of the assembly. It may be a shim, screw, nest, collet.

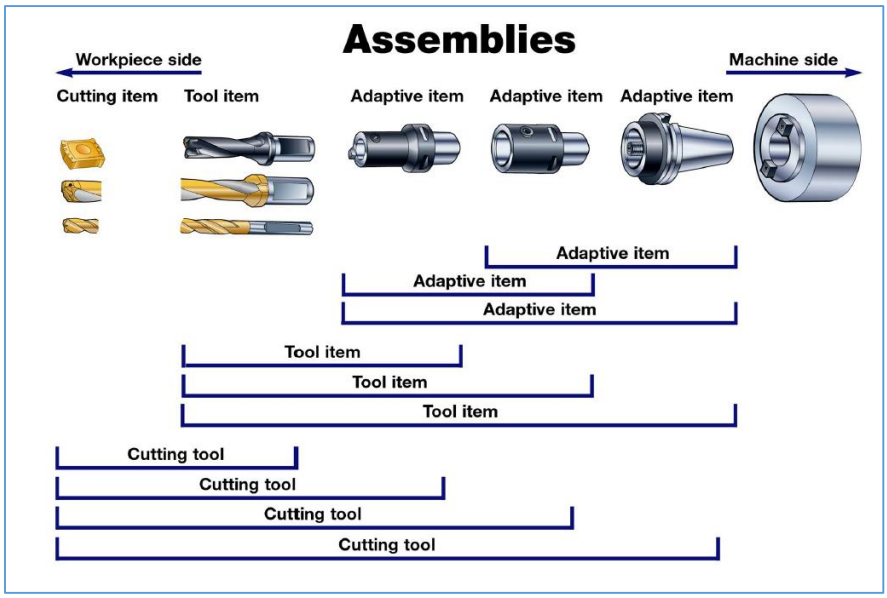

Figure 1: Items with reference to workpiece and machine side [1]

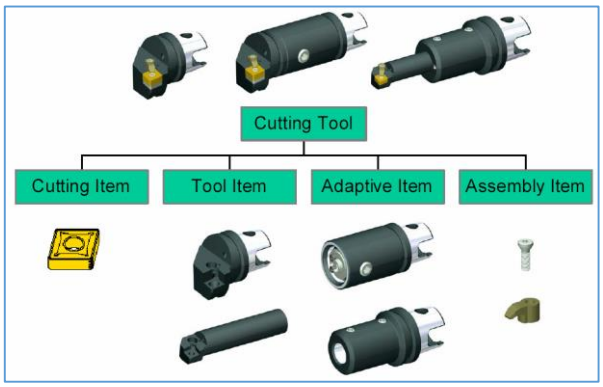

Figure 2: Information about these groups is exchanged [1]

3.1.2 Product file

A p-21 file is composed of:

Header Part: ISO part, keywords and entity instances must be defined according to the header section in example. The entity instance *FILE_SCHEMA() with the argument 'CUTTING_TOOL_-SCHEMA_ARM'* determines that the content of this file meets the definitions and rules of the following EXPRESS schema: Application Reference Model ISO 13399-1, Annex C in the context of cutting tools.

Data Part: Each line with a unique number (#number > 0) is an entity instance. EXPRESS provides simple and complex entity data types. Every entity data type in EXPRESS corresponds to the definition of an application object in ISO 13399-1. Each entity instance can be referred by other entity instances as an attribute value, whereby it is regardless of whether the entity instance is defined before or after using it as an attribute.

3.2 ISO 13399: Main concepts

ISO 13399 is a standard for cutting tool data representation and exchange. Figure 3 shows different aspects in the context of ISO 13399. The ISO 13399 documents as well as parts library for cutting tools and product file p21 are the implementation relevant issues. The ISO 13399 standard was mainly developed for the purpose of enabling the digital tool data exchange between and inside companies and for implementing and sharing cutting tool product data. It was developed with contributions from major world cutting tool vendors and research institutes. The schema of cutting tool data representation is defined in ISO 13399-1 and its usage guidelines in ISO 13399-150. If the relations in the schema are applied to the product data according to the usage guidelines, the resulting p21-file is sufficiently general to be readable for other systems.

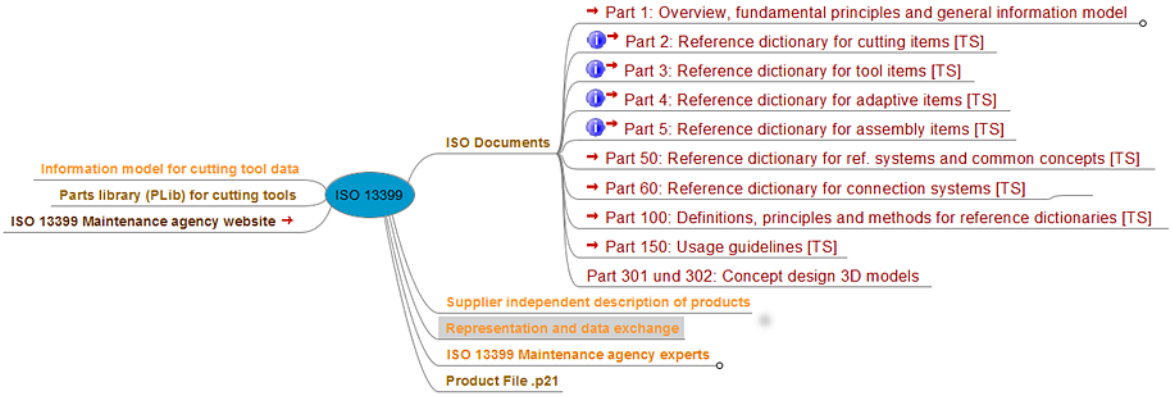

Figure 3: Different aspects of ISO 13399

3.2.1 Information model and ontology

ISO 13399 defines **a generally applicable information representation model** which is **used together with an explicit reference dictionary**. ISO 13399 can also be seen as an ontology, as it is interpreted in the field of information science: as an explicit specification of a conceptualization [D_11]. This specification is in document ISO 13399-1. In the next chapters 3.2.2 – 3.2.5 some of the specified concepts are introduced.

The reference dictionary is a parts library, which is the formal representation of an ontology and provides item and feature classes as well as assigned properties. In the remainder of this document the concrete parts library file for cutting tools is called PLIB-file.

3.2.2 Industrial automation systems and integration - parts library

The explicit reference dictionary used in ISO 13399 is an “Industrial automation systems and integration parts library” and is defined in ISO 13584-25. ISO 13584 was initially created to support electronic catalogs of components, but the scope has later been widened to support the creation and electronic representation of a dictionary. The concrete PLIB-file version is released by the ISO 13399 maintenance agency [PLib]. An advantage of the separation into schema and parts library is the possibility to update the PLIB – file and keep the schema untouched.

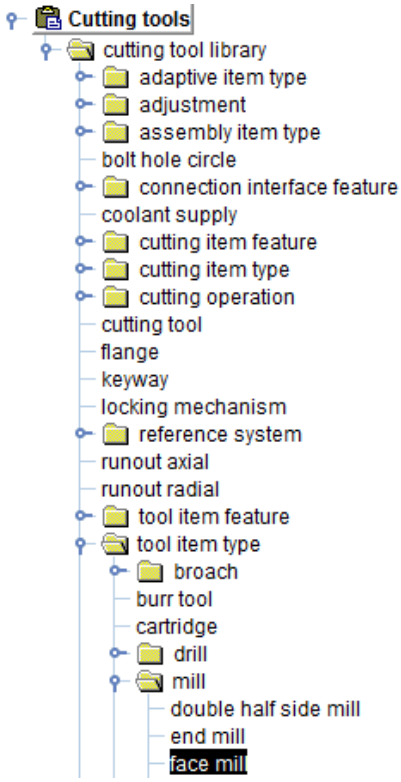

Figure 4: Classification for a face mill in PLib

Figure 4 shows the classification for a face mill in the Plib. The ISO committee is taking in consideration a structure revision. The idea is to consider parts of the GTC hierarchy in the class structure of the parts library. Decisions of a revised structure will be taken at the earliest in April 2015. Considering these eventualities the description of the parts library is limited to main mechanism.

Coding mechanism: In the GTC Guidelines [D_12] the PLib is referred to as a "decoder" file (see Figure 6). This statement has to be interpreted as follows: A p21-file contains a PLIB_CLASS_REFERENCE – entity, which is a link to a class in the PLIB-file. Only if the PLib version '002' with the supplier_bsu '0112/1///13399' is used to interpret the p21-file, the class behind the code is recognized as the targeted class. Each basic semantic unit, either a class or a property, is identified with a unique code ('71D1AA6635E76').

```
ENTITY PLib_class_reference;
code : STRING;
supplier_bsu : STRING;
version : STRING;
END_ENTITY;
```

```
ENTITY PLib_property_reference;
code : STRING;
name_scope : PLib_class_reference;
version : STRING;
END_ENTITY;
```

EXPRESS Specification 1: PLIB-references

Explicit classes and properties: Figure 5 shows the item class *end mill*, which has a name, a code, a definition and is described by a note. A set of properties is assigned as applicable on the class and one of them is *body length*. A property also has name, code, definition and note. In addition the property has a data type, which is *real measure type* for the property *body length*.

Relations: The class hierarchy in the PLIB is very flat and extensive subclassing was avoided. All properties are defined as visible on the item class *cutting tool library* and are assigned to the classes as applicable, where they can be meaningfully applied. This approach has the advantage that for instance in the case of a tool with multiple functions properties can be assigned without class structure changes.

Kind of classes: Two different class types exist in PLIB. ITEM_CLASSES correspond to the different items in chapter 3.1.1. The class *end mill* shown in Figure 5 is an ITEM_CLASS because it is a subclass of *tool item type*. The class *bolt hole circle* in Figure 7 is a FEATURE_CLASS, which in the context of ISO 13399 behaves different from ITEM_CLASSES. That means it collects properties related to the feature [D_11].

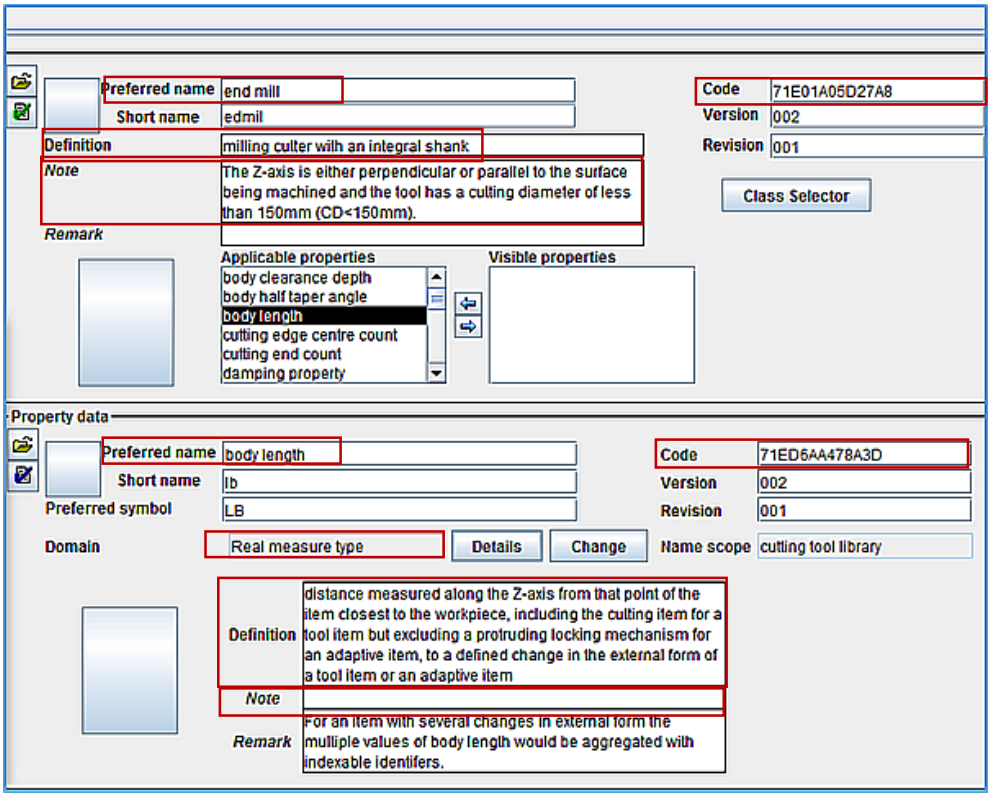

Figure 5: Explicitly defined class and property [displayed by PLIB Editor]

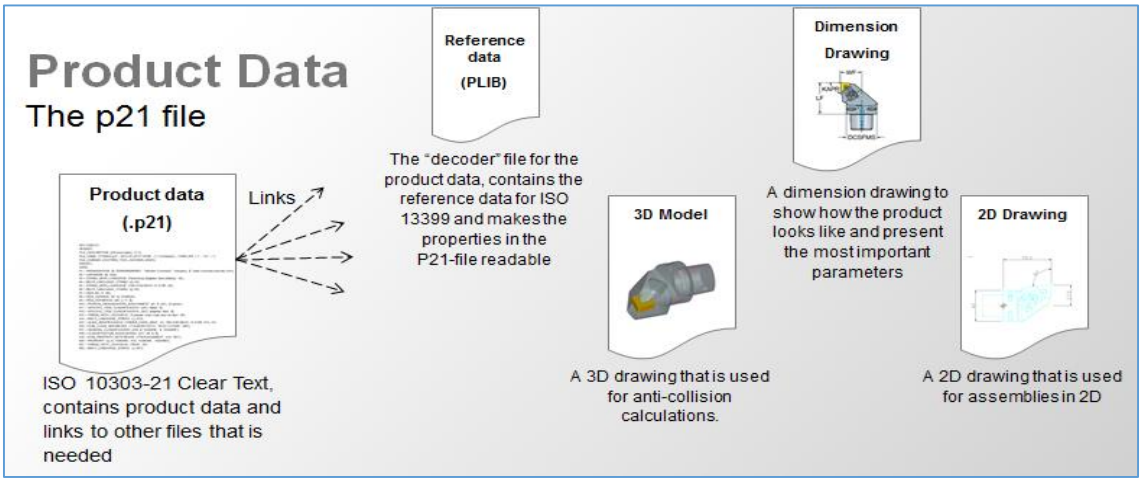

Figure 6: From GTC Guidelines [D_12]

Applicable and visible properties: The set of applicable properties assigned to a class characterizes the class. Figure 7 shows the feature class *bolt hole circle*. The applicable properties b*olt hole circle count, diameter access hole* and *diameter bolt circle* specify the concrete form of the feature bolt hole circle. The visible property *clamping length maximum* is not an essential characteristic of the form of this feature. In general an applicable property is meaningful in the scope of a given characterization. But the same property is not necessarily applied to a product belonging to this class. In this case the property has the status visible.

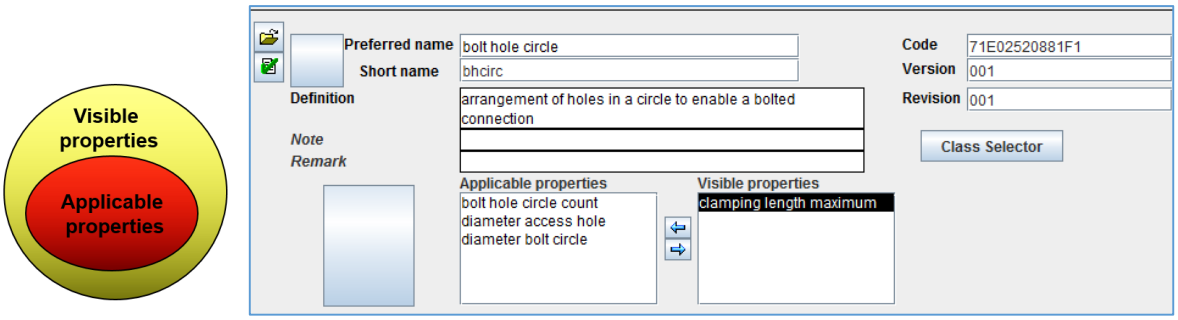

Figure 7: class bolt hole circle with applicable and visible properties

3.2.3 Units of Functionality - Concept

The information requirements of ISO 13399 standard covers different "Units of Functionality", which use defined Application objects. Such a unit of functionality is *classification*, which allows to classify items as materials, parts or tools. Items can additionally be classified by their attributes as screws, bolts, nuts, shafts or brackets. Other units of functionality concern the representation of geometric dimensions/tolerances or different extern reference mechanism. Chapter 5 of ISO 13399-1 contains the detailed description of "Units of Functionality".

3.2.4 Application objects

All application objects are defined within schema ISO 13399-1. As shown in Figure 8 the content of an application object is the EXPRESS specification. Every entity instance in a p21-file corresponds to such an EXPRESS definition (see product file description, data part).

3.2.5 Reference mechanism

The reference mechanism is another functionality in ISO 13399. It allows to reference for instance to classes and properties in a PLIB, to external documents or to properties in DIN 4000 standard.

PLIB references

Application object *PLib_class_reference:* #64 = PLIB_CLASS_REFERENCE ('71D1AA6635E76', '0112/1///13399', '002'); Entity instance #64 references an explicit class with code '71D1AA6635E76' in PLIB version '002' with supplier_bsu '0112/1///13399'.

Application object *PLib_property_reference*: #67 = PLIB_PROPERTY_REFERENCE ('71CF29872F0AB', #64, '001'); Entity instance #67 references an explicit property with code '71CF29872F0AB' in PLIB version '001' and in the scope of the class referenced in entity #64.

External References

Application object *external_library_reference:* #74 = EXTERNAL_LIBRARY_REFERENCE ($, 'PRODDESCR', 'CSC'); Entity instance #74 references an external library with empty description, external id 'PRODDESCR' and vendor specific library 'CSC'.

Application object *external_file_id_and_location*: #54 = EXTERNAL_FILE_ID_AND_LOCATION ('112321.JPG', #53); Entity instance #54 references an external file with the external id '112321.JPG' and the location defined in Entity instance #53.

Other application objects which provide external references are: digital_document, document or external_geometric_model.

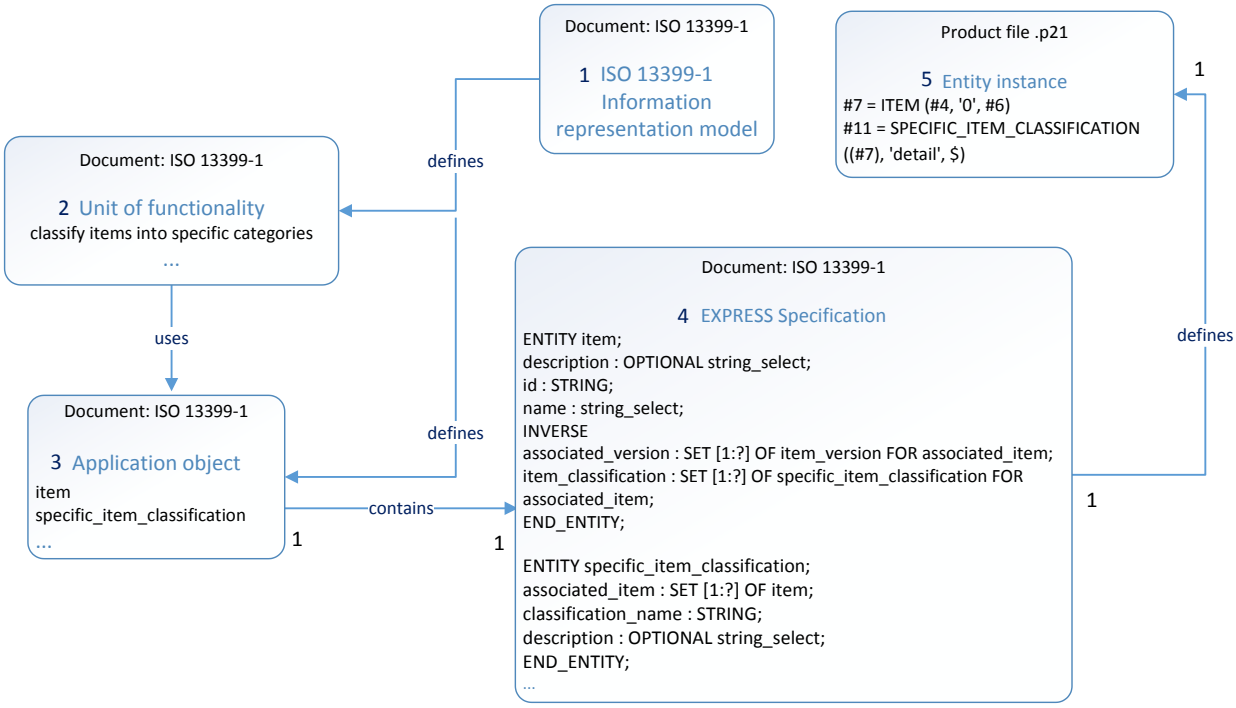

Figure 8: Definitions and instances

4 GTC (Generic Tool Catalog)

The meaning of GTC is “Generic Tool Catalog”. The GTC format covers on one hand the “GTC hierarchy”, a classification for product variants and on the other hand the “GTC package”. The GTC package typically includes information on GTC hierarchy.

The following terminology explains the most important terms in the context of GTC:

GTC package: digital product catalog delivered in “GTC format”.

GTC format: defined structure and content of a digital product catalog

GTC package specification: separate document where structure and content for a digital product catalog is described in detail; download on GTC website.

GTC format implementation: All the work that a company has to do to be able to exchange cutting tool product data in “GTC format”. This covers data classification and application development for to read and write product data in “GTC format”.

GTC hierarchy: classification hierarchy

The purpose and motivation to develop and introduce GTC as a complement to ISO 13399 standard is to smooth catalog data exchange. The two main purposes are:

1. It allows applications to import digital catalog data and map it correctly into a standardized hierarchy.
2. It allows vendors to provide their catalogs in a standardized structure

The link between GTC hierarchy and GTC package is the GCT class id, which is explained in chapter 4.1.2. The GCT class id is a mandatory field within the assortment file.

4.1 GTC hierarchy

The GTC hierarchy consists of generic and vendor specific areas and of fixed and rule based levels.

Fixed levels are maintained by the GTC governance organization and are provided on the GTC website. For each class a mapping rule specifies the criteria for selecting products belonging to this class.

Rule based levels come after the fixed levels and are affected by the assortment each tool vendor has. GTC governance organization provides abstract classes and mapping rules on these levels 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 {titp}
	Adaptive item	select {ait}
	Rotation symmetrical adaptor	select [RPMX]>0 AND NOT {collet}
	Collet	select {collet}
Abstract classes in rule based level	Machine side [connection interface]	define {connect_mach}
	Workpiece side [connection interface]	define {connect_wkps}

Figure 9: classes in fixed and rule based levels

4.1.1 Mapping rules

The mapping rules are defined on an abstract level. For applying them in a machine readable form, the mapping logic has to be implemented in the individual data management system of a company. This chapter provides an explication of the abstract rules and their links to different kinds of PLib class and to Plib properties.

Notational convention

{} Represents an ISO 13399 PLib class, enclosed text is a class short name.

[] Represents an ISO 13399 property, enclosed text is a property symbol.

The following examples point out distinctions among rules, based on different expressions used in the rule. The context is use case 1, so product variants are chosen by means of mapping rules.

1. Every rule is either a **select** rule or a **define** rule.

    a. **Select rule**: Used as filter applied on a set of items which result from parent filters

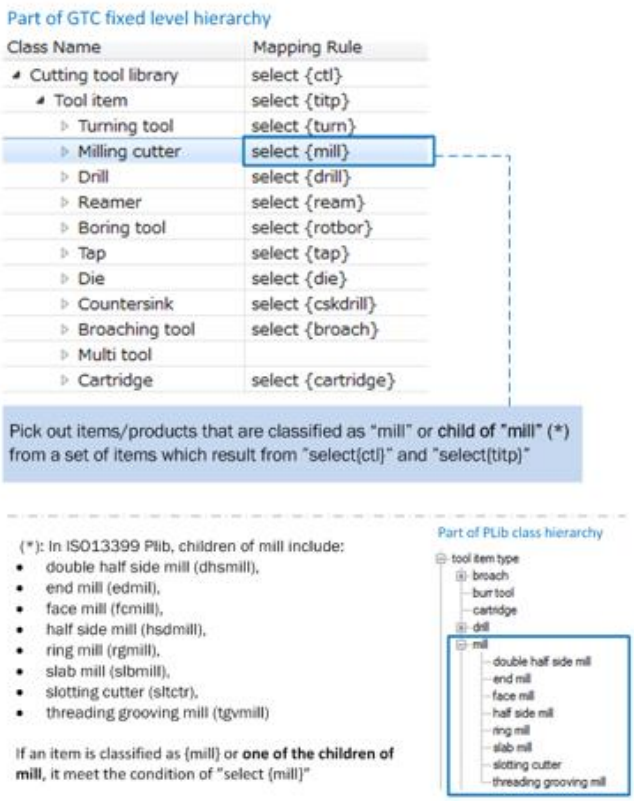

Figure 11: Example of a ‚select' rule

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

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 12: Example of rules based on connection interface

2. Rules using a Plib FEATURE_CLASS in expression

Use case 1, group products by their connection interface features, see Figure 12

Define {connect_mach}

FEATURE_CLASS 'connection interface feature' with the additional value 'mach' for property [SIDE]

take products which have the same **machine** side connection interface into one group

Define {connect_wkps}

FEATURE_CLASS 'connection interface feature' with the additional value 'wkps' for property [SIDE]

take products which have the same **workpiece** side connection interface into one group

3. Rules using one or more PLib ITEM_CLASSES in expression

select {mill}

uses ITEM_CLASS 'mill'

filter products which are classified as 'mill' or as child of 'mill'

select {trnint} AND (NOT {trnext})

uses two ITEM_CLASSES: 'turning internal' and 'turning external', which are 'cutting operation' classes

filter products of cutting operation 'turning internal'

select {minst} or *select NOT {minst}*

uses ITEM_CLASS 'master insert', which is a 'reference system' class

felect {minst}: filter products of reference system 'master insert', that means indexables
select NOT {minst}: filter products that are solid tool items

4. Rules using PLib PROPERTY in expression

select [RHO]{fdp}=180

uses PROPERTY 'RHO' in the scope of ITEM_CLASS 'feed direction primary', which is a 'reference system' class

filter products, where the PROPERTY 'RHO' in the scope of ITEM_CLASS 'feed direction primary' has the value 180

4.1.2 GTC class id

The GTC class id is a string which contains the information to identify the class a product belongs to. The GTC class id is a mandatory field in the assortment file and the assortment file in turn is a mandatory file of the GTC package in the case of including product data. The following examples show how the GTC class id is composed of different parts separated by underscores.

Class Name	Class ID
Cutting tool library	CTL
Tool item	TL
Milling cutter	MIL
Face milling cutter	MILF
Shoulder face mill	MILSQ
indexable	MILSQI
[insert concept]	MILSQI_W[insert_concept]
[connection interface]	MILSQI_W[insert_concept]_M[connect_mach]

Figure 13: GTC hierarchy with class id

Example 1: Tool item with ISO insert concept

MILSQI_ISO$S_FDA12 ClassesInFixedLevel_InsertConcept_MachineSideConnectionInterface

ClassesInFixedLevel: Put class id strings together from start level 'child of tool item' until last of fixed levels according to Figure 13

InsertConcept**:** Put string together according to ISO 13399-60 [chapter 5.2]:
ISO$: four-digit string for ISO-designated insert + S: Insert type = square = ISO$S

MachineSideConnectionInterface: Put string together according to ISO 13399-60 [x]:
FDA: short name of connection FEATURE_CLASS 'FDA milling arbor connection' + 12: variant 12 = FDA12

Example 2: Tool item with 'MB' insert concept

MILSQI_MB$$XX_FDA22 ClassesInFixedLevel_InsertConcept_MachineSideConnectionInterface

ClassesInFixedLevel: see example 1

InsertConcept**:** Put string together according to ISO 13399-60 [chapter 5.2]:
MB$$: four-digit string for ISO-designed name of manufacturer 'MB'[1] + XX: specific insert of manufacturer 'MB'

MachineSideConnectionInterface: Put string together according to ISO 13399-60 [x]:
FDA: short name of connection FEATURE_CLASS 'FDA milling arbor connection' + 22: variant 22 = FDA22

Example 3: Adaptive item

ADPCL_MZYL11_WZYL01 ClassesInFixedLevel_MachineSideConnectionInterface_ WorkpieceSideConnectionInterface

ADPCL: Put class id strings together from start level 'child of adaptive item' until last of fixed levels according to Figure 12

M: SIDE = 'machine' + ZYL: short name of connection FEATURE_CLASS 'ZYL cylindrical shank connection' + 11: variant 11 = MZYL11

[1] fictive company

W: SIDE = 'workpiece' + ZYL: short name of connection FEATURE_CLASS 'ZYL cylindrical shank connection' + 01: variant 01 = WZYL01

4.2 GTC package

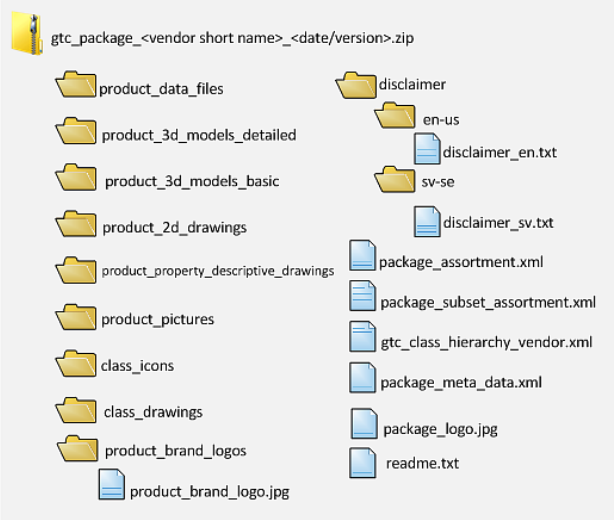

Figure 14: GTC package folder structure

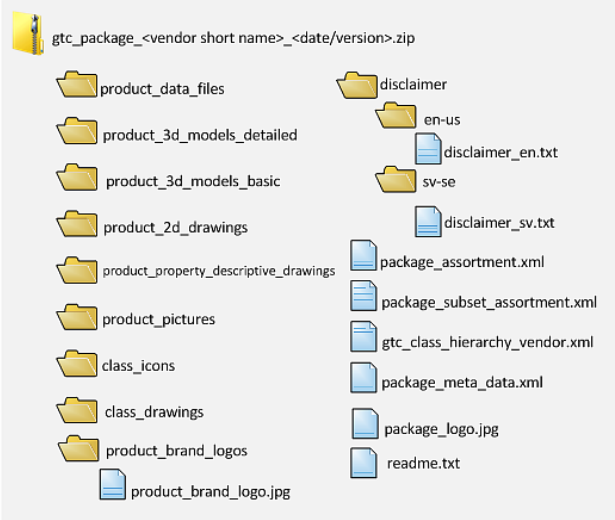

Figure 14 shows the GTC package folder structure as it is defined in the „GTC package specification“ and as digital product catalogs are expected to be delivered by cutting tool vendors. The specification document will be provided on the GTC website and defines – besides the folder structure - which files and file contents are mandatory and which file formats are allowed or recommended. This chapter sets the focus on the files which the cutting tool vendor provide according to their individual assortment.

4.2.1 Overview GTC package content

Table 1 the main content for the GTC package - provided by a tool vendor - is listed.

Data package			GTC hierarchy
Assortment file	Product File	3D Model, 2D drawing, pictures	GTC vendor hierarchy, class drawings, class icons
package_assortment.xml - Products classified according to GTC classification hierarchy,	.p21 files in folder product_data_files 1 file for each product	STEP 3D models as .stp files in folder product_3d_models basic or detailed	gtc_class_hierarchy_vendor.xml based on the GTC generic hierarchy and the vendor's products

GTC class id for every product - Information of data changes - Information of lifecycle changes	Items with - Applicable properties - Associations - Class definitions - References to external libraries	2D drawings as .dxf files in folder product_2d_ drawings Product drawings in folder product_property_ descriptive_drawings Product Pictures in folder product_pictures	class drawings showing important properties in each GTC leaf node class -> folder class_drawings icon pictures of GTC class nodes -> folder class_icons

Table 1: The main files delivered by the tool vendor

4.2.2 Assortment file

The assortment file has great importance for the effective product data management. By the means of the assortment file, the vendor provides a useful overview of the supplied data within the digital catalog.

Product_id	P21_file_name	Order_code	Unit_system	GTC_class_id	GTC_status	P21_status
5724529	5724529.p21	A393.T-19 10 175	Inch	ADPRS_MZYL01_WSAC03	2012-12-07, 00:00:00	2013-01-07, 00:00
5844001	5844001.p21	A490-125J38.1-14M	Metric	MILSQI_SV$$MI_FDA12	2012-12-07, 00:00:00	2013-01-07, 00:00
5721215	5721215.p21	AA3B20-50 16 197	Inch	ADPRS_MSKG37_WZYL10	2012-12-07, 00:00:00	2013-01-07, 00:00
5728856	5728856.p21	C6-391.CGB-12 092A	Metric	ADPRS_MCCS01_WZYL01	2012-12-07, 00:00:00	2013-01-07, 00:00
5729140	5729140.p21	C8-391.0204-63 090	Metric	ADPRS_MCCS01_WCCS04	2012-12-07, 00:00:00	2013-01-07, 00:00
6228924	6228924.p21	C8-A391.05C-19 030M	Inch	ADPRS_MCCS01_WFDA12	2012-12-07, 00:00:00	2013-01-07, 00:00
6181422	6181422.p21	E003M5	Metric	TAPCCYLF_ZYL21	2012-12-07, 00:00:00	2013-01-07, 00:00
6181073	6181073.p21	E011M6X.5	Metric	TAPCCYLG_ZYL21	2012-12-07, 00:00:00	2013-01-07, 00:00
6182098	6182098.p21	E0227/8	Metric	TAPCCYLF_ZYL21	2012-12-07, 00:00:00	2013-01-07, 00:00
6181813	6181813.p21	E0305/8	Metric	TAPCCYLG_ZYL21	2012-12-07, 00:00:00	2013-01-07, 00:00
6181248	6181248.p21	E0389/16	Metric	TAPCCYLF_ZYL21	2012-12-07, 00:00:00	2013-01-07, 00:00
6182879	6182879.p21	E0571/2	Metric	TAPCCYLG_ZYL21	2012-12-07, 00:00:00	2013-01-07, 00:00
6181961	6181961.p21	E091M3	Metric	TAPFCYL_ZYL21	2012-12-07, 00:00:00	2013-01-07, 00:00
5733663	5733663.p21	E25-A32-SS-080	Metric	ADPRS_MZYL01_WSAT01	2012-12-07, 00:00:00	2013-01-07, 00:00
6182079	6182079.p21	E308M4	Metric	TAPFCYL_ZYL21	2012-12-07, 00:00:00	2013-01-07, 00:00
6182250	6182250.p21	E347M27	Metric	TAPCCYLF_ZYL21	2012-12-07, 00:00:00	2013-01-07, 00:00
6181157	6181157.p21	EX10M20X1.5	Metric	TAPCCYLF_ZYL21	2012-12-07, 00:00:00	2013-01-07, 00:00
6181982	6181982.p21	EX211/4	Metric	TAPCCYLF_ZYL21	2012-12-07, 00:00:00	2013-01-07, 00:00
6181597	6181597.p21	EX411	Metric	TAPCCYLF_ZYL21	2012-12-07, 00:00:00	2013-01-07, 00:00

Figure 15: Extract from assortment file

The assortment file contains – amongst others – information about:

1. which GTC class a product belongs to
2. where hierarchy changes have taken place:
    a. current GTC version number
    b. new, moved or removed nodes
    c. at which nodes the property list changed
3. where EXPRESS schema changes or STEP entity usage changes have taken place
4. value changes and life cycle changes concerning products

With the detailed information for every product the receiving system can find out in advance which product file have to be processed. For complete information see document “GTC package specification” [FD_1].

4.3 GTC generic content for fixed hierarchy levels

In order to implement the GTC format, tool vendors and application developers need the GTC generic content, which is provided on the GTC website. In Table 2 the administrated contents and their purpose is described.

Content	**Purpose**
GTC classification hierarchy fixed levels	base for classification of vendor specific products (gtc_class_hierarchy_vendor.xml)
list of **relevant properties**	Know relevant mandatory and optional properties of Plib (and other external libraries) for the class a product belongs to: - to capture new products on vendor specific node - to know what must be read from a p21-file and where it can be found
class drawings	If products within the GTC hierarchy are displayed in an application, there are defined drawings to display for classes on fixed levels
icon pictures	If products within the GTC hierarchy are displayed in an application, there are defined icons to display for classes on fixed levels

Table 2: GTC generic content provided on GTC website

5 GTC implementation road

There are two main goals which can be met if the GTC format is implemented. On one hand the format of a single cutting tool vendor's product data can be understood by all receiving systems. And on the other hand a receiving system is able to read the product data of different vendors with the same application respectively the same code. So the implementation road includes all steps that are necessary to provide as well as receive product data catalogs in GTC format as shown in Figure 16.

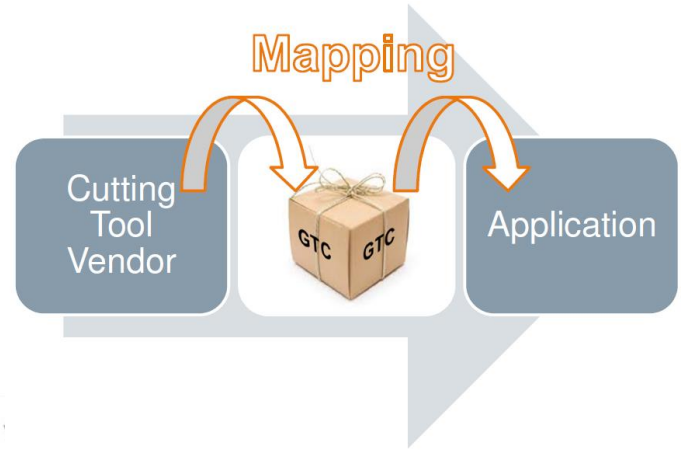

Figure 16: from GTC day introduction

5.1 Tool vendor road

Figure 17 illustrates the implementation steps for the vendor side. The goal is to deliver the GTC package of the vendor specific assortment for various queries.

1. The mapping of the product properties to ISO 13399 Plib properties and the creation of the p21 file - under the condition that the ISO 13399 schema is met - have to be implemented. This is the guarantee that the product files can be read by applications which expect product data in ISO 13399 standard.

2. The link from vendor specific to GTC classes has to be implemented. Therefore vendor specific classes may first be mapped to or even replaced by ISO 13999 classes. Then the mapping to GTC classes can be implemented and the GTC class hierarchy with the vendor specific products can be created.

3. Product files, 3D models, 2D drawings, icons **and all demanded additional files and pictures** according to the GTC package specification [FD_1] have to be created and provided together with the product data.

4. To complete the GTC package the creation of the **assortment file** and the **metadata files** according to GTC package specification [FD_1] has to be implemented.

5. The digital catalog in GTC format has to be provided **for different queries**: full catalog, only changed product data for update or only for GTC hierarchy information without product data.

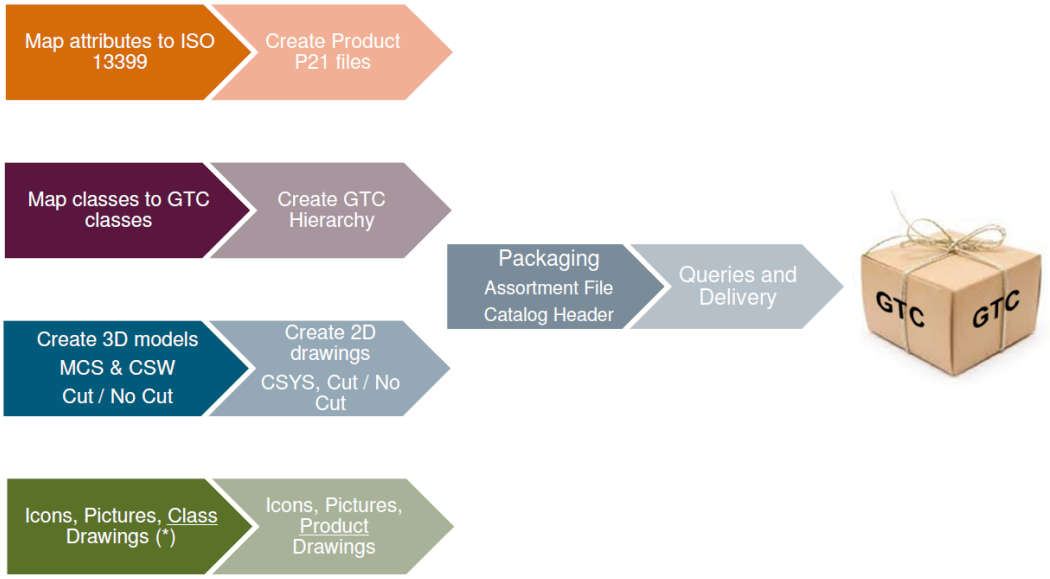

Figure 17: from GTC day introduction

5.2 Application road

Figure 18 illustrates the implementation steps for the application side. The goal is to unpack the received GTC package and create or update the catalog in GTC format. Then the received product data can be displayed in the context of the GTC hierarchy.

1. The **unpacking and data processing** of received digital catalogs in GTC format has to be implemented.

2. The creation and update of the GTC hierarchy has to be implemented for **updating the GTC classes** with the received GTC class hierarchy information. This could be for example a new class on the fixed levels which corresponds to use case 5.

3. The application side has to implement the **mapping of ISO 13399 attributes** to the receiving data system. This implementation allows to read the p21 files and update the products.

4. The processing of 3D models, coordinate systems, Cut/No Cut information as well as of drawings and pictures has to implemented, so the models and drawings in the receiving system can be updated.

5. The **processing of the assortment file** has to be implemented, which means to implement the removing or adding of product variants as well as the update of product life cycle information.

Remark: If on the application side the rule based levels of the GTC hierarchy are not implemented, there is a possibility to read features and properties from the p21 file into the receiving data system.

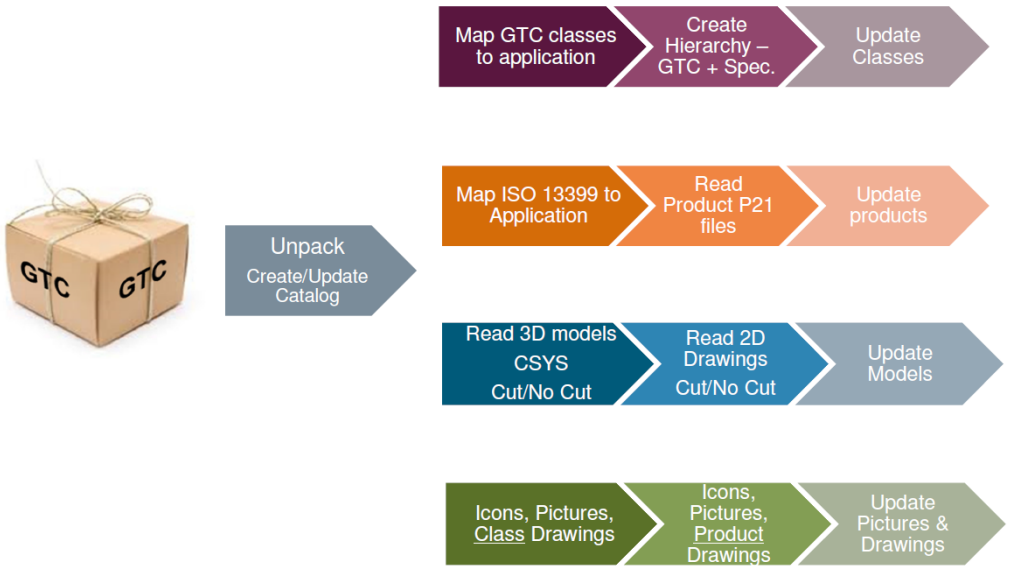

Figure 18: from GTC day introduction

6 Appendix

6.1 Further documents for GTC implementation

Name	Author	Download Location	Reference
GTC package specification		Download	[FD_1]
GTC mapping rule specification		Download	[FD_2]

6.2 Bibliography

ISO 13399 documents	Download Location, feebased
Part 1: Overview, fundamental principles and general information model	Part1
Part2	Part2
Part3	Part3
Part4	Part4

Part5	Part5
Part50	Part50
Part60	Part60
Part100	Part100
Part150	Part_150
Part301+302	Part 301 Part 302

Title	Content	Author	Download Location	Reference
Information Management for Cutting Tools	Doctoral Thesis, KTH, Stockholm, 2008	Olof Nyqvist	Download [D_11]	[D_11]
GTC Guidelines	Sandvik Coromant Technical White Paper	Bengt Kinnvall, Bengt Olsson	Download [D_12]	[D_12]
EXCHANGE OF COMPONENT DATA:THE PLIB (ISO 13584) MODEL, STANDARD AND TOOLS	Proceedings of the CALS EUROPE'98 Conference, 16-18 September 1998	G. Pierra and E. Sardet, LISI/ENSMA et. Al.	Download [D_13]	[D_13]
PLIB Modeling: Part 3 Principles of classification using ISO 13584	ISO TC184/SC4 Vice Equines Meeting in Italy, 2006	Hiroshi MURAYAMA	Download [D_14]	[D_14]

6.3 Tools and content

Name	Function / content	Author	Download Location	Reference
PLIB Editor	Display and edit PLIB-file (.spf or .p21)	ENSMA École nationale supérieure de mécanique et d'aérotechnique	Download [T_1]	[T_1]
PLIB	Classes and properties as defined in ISO 13399 documents Part 2-5	ISO 13399 maintenance agency	Download [T_2]	[T_2]

6.4 Use cases

The following use cases show situations either on the vendor or application side.

Vendor side:

Use case 1: The vendor generates the GTC rule based level hierarchy with his product variants. The resulting file is the package_assortment.xml in the GTC package root folder.

Use case 2: The vendor generates the assortment file including the GTC class id for his product variants for the GTC package.

Application side:

Use case 3: The receiving system reads a full digital catalog in GTC format.

Use case 4: The receiving system reads data of new or changed products. --> New version of digital vendor catalog in GTC format with product data.

Use case 5: The receiving system reads changed GTC hierarchy information of fixed levels. --> New version of GTC hierarchy / new GTC release.

Use case 6: The receiving system reads changed GTC hierarchy information of rule based levels. --> New version of digital vendor catalog in GTC format with hierarchy information.

6.5 Extracts from doctoral thesis

Olof Nyqvist proposed two hypothesis in his doctoral thesis “Information Management for Cutting Tools” [D_10]:

Hypothesis 1: It is possible to create an information representation for cutting tools that is both generic and explicit.

Hypothesis 2: A generic information representation for cutting tools requires reference data in order to be explicit.

In the doctoral thesis the different items are defined as follows:

> A cutting item is either an insert or the cutting edge of a solid tool, it holds information about the cutting portion of a tool.
>
> A tool item holds a cutting item. It may be the tool body of an assembly or the body of a solid tool.
>
> An adaptive item modi¯es the properties of an assembly and cannot hold a cutting item by itself, e.g. an adaptor or extender.
>
> An assembly item is a component necessary for the complete assembly of a tool. It may change properties of components but does not change the properties of the assembly. It may be a shim, screw, nest, collet etc.