ISO ToolMaker application development

Darya Botkina & Meixin Xu

Master’s thesis project,

Stockholm, Sweden 2016
Abstract

With the rapid development of manufacturing industry, there is an increasing need for different industries to communicate, exchange and share information. While, the diversity of terms being used among different organizations makes the communication hard to proceed. Due to this reason, Sandvik Coromant decided to initiate development of a new international standard ISO 13399 under the title Cutting tool data representation and exchange. The standard is developed by Sandvik Coromant, KTH Royal Institute of Technology in Stockholm, French Cetim (technical centre for mechanical engineers) and other organizations in the metal cutting sector. (Sandvik Coromant)

ISO 13399 has been developed for years, and it has reached a stage to be tested and verified. In 2014, Sandvik starts a new project of developing an application called ISO ToolMaker for verifying and validating the development and implementation specification of ISO 13399. By 2016, the framework of this application has been set up. Since the software development is still at the primary stage, there are plenty of issues need to be settled afterwards.

Therefore, the purpose of this thesis is to promote the reform of ISO ToolMaker, adjusting existing undesired functions, adding new modules for advanced usage, and reorganize the software structure for continuous improvement. By conducting the literature research and interviewing the staff in Sandvik Coromant, an action plan was made, and a systematic method was applied to guide the procedures.

Compare to the previous application, the final version has three significant advantages - an extra store function for items and assemblies, a more rigorous logical connection and a one-click output mode and, in addition, the text-info for guiding each step and the detail design of the software. All the aspects make it a superior and user-friendly application to work with.
Sammanfattning

Med den snabba utvecklingen av industrin finns det ökande behov mellan branscherna att kommunicera, utbyta och dela information. Samtidigt försvarar mångfalden av termer som används mellan olika organisationer kommunikationen. På grund av detta beslutade Sandvik Coromant att initiera utveckling av en ny internationell standard ISO 13399 med titeln Cutting tool data representation and exchange. Standarden är framtagen av Sandvik Coromant, Kungliga Tekniska högskolan i Stockholm, franska CETIM (tekniskt centrum för maskiningenjörer) och andra organisationer i metallbearbetningssektorn. (Sandvik Coromant)


Därför är syftet med detta examensarbete att vidareutveckla ISO ToolMaker, genom att förbättra dåliga funktioner, lägga till nya moduler för avancerad användning och omorganisera mjukvarustrukturen för ständiga förbättringar. Som ett resultat av litteraturstudie och intervjuer med personal på Sandvik Coromant togs en handlingsplan fram och en systematisk metod användes för utvecklingsarbetet.

Jämför med det tidigare verktyget, har den slutliga versionen tre betydande fördelar - en extra lagringsfunktion för objekt och montage, en mer omfattande logisk sammankoppling och en ”one-click”-resultatvisning och dessutom textinfo för att styra varje steg och detaljutvecklingen av mjukvaran. Dessa egenskaper gör det till ett bekvämt och användarvänligt verktyg att arbeta med.
Acknowledgements

This thesis project is held by both KTH Royal Institute of Technology and Sandvik Coromant in Sweden. First, we want to thank our supervisor Thomas Lundholm for introducing us to this amazing project in Sandvik and for his endless help and support during the whole work period. We want to send our special thanks to Bengt Olsson for providing us this topic and coming all the way from Sandviken to Stockholm so many times to patiently answer our questions and offer help.

Many thanks to Boyang Li and Denys Knertser for their great assistance and support during our Java learning process and always being so helpful when we encounter problems; Thanks to Julie Huang for taking her time to introduce the situation and provide the useful information of this project to us; and thanks to Ove Bayard for all the thoughtful arrangement for the presentation and defense opportunity in such a tense time.

In the end, we would like to thank all our parents, families and friends who have always been so supportive and been on our side no matter what happens, and hope these sincere friendships and love last a lifetime.

Stockholm, October 2016

Darya Botkina & Meixin Xu
Table of Contents

1. Introduction ................................................................................................................................. 1
   1.1 Background and literature study ........................................................................................ 1
      1.1.1 Background .............................................................................................................. 1
      1.1.2 ISO 13399 ............................................................................................................... 1
      1.1.3 Communication requirement .................................................................................... 3
      1.1.4 Communication tool ............................................................................................... 4
      1.1.5 Communication benefit .......................................................................................... 5
      1.1.6 ISO ToolMaker ....................................................................................................... 5
   1.2 Purpose ................................................................................................................................... 5
   1.3 Main features of the information model ............................................................................. 6
      1.3.1 Different types of items ............................................................................................. 6
      1.3.2 Applied example ...................................................................................................... 8

2. Method ...................................................................................................................................... 10
   2.1 Problem solving methodology ............................................................................................. 10
   2.2 5-step approach application ............................................................................................... 11
   2.3 Optimization goals ............................................................................................................. 13

3. Analysis and result .................................................................................................................... 15
   3.1 Assembly compilation ........................................................................................................ 15
   3.2 ToolMaker 1.0 .................................................................................................................. 17
   3.3 ToolMaker 2.0 .................................................................................................................. 20
      3.3.1 Algorithm .................................................................................................................. 21
      3.3.2 ToolMaker 2.0 interface .......................................................................................... 22
      3.3.3 Renaming of the assembly ....................................................................................... 27
      3.3.4 Export of the assembly ............................................................................................ 28
      3.3.5 Editing of the assembly ........................................................................................... 28
   3.4 My Items and My Assemblies interfaces ............................................................................ 30

4. Further work and development ............................................................................................... 37
   4.1 3D models file management ............................................................................................. 37
   4.2 Additional export ............................................................................................................... 37
   4.3 AdveonTM ......................................................................................................................... 37
   4.4 Tweeting machine .............................................................................................................. 38

5. Bibliography ............................................................................................................................. 39
1. Introduction

This chapter provides the introduction of this thesis project to help readers establish a better initial understanding about the background, the purpose, and the related information etc. After reading this chapter, readers should be able to know the big environment behind this project, how the thesis work contributes to the big picture, and the practical sample being used in this case.

1.1 Background and literature study

1.1.1 Background

Sandvik Coromant is the world leading supplier of tools, tooling solutions and material technology for the metalworking industry. As an industry leader, Sandvik offers various products and solutions for their customers. The applied advanced technology not only benefits all their customers, but also drives the entire industrial forward.

As the development of the whole manufacturing industry, a new concept digital plant is brought forward, which is changing the entire industrial landscape. Digital plant is evolved to achieve true digital manufacturing. It makes production possible to develop, process, simulate, and test digitally, thus vastly reducing the requirement for physical prototyping and waste produced during production. (Nyqvist, 2008)

Cutting tools play an important role in digital manufacturing, and there is a need for standardized cutting tool data. Besides, there is an increasing demand for different industries to communicate, which includes exchange data, and share information between or within companies. While, the diversity of used terms among suppliers and customers makes it complicated to gather and convert data. A way to connect all the parties together and freely share information seems urgent.

For all these conditions, Sandvik decided to set a new standard to solve the communication problem and initiated the development of the international standard ISO 13399 under the title Cutting tool data representation and exchange. This standard is being developed jointly by Sandvik Coromant, KTH Royal Institute of Technology in Stockholm, French Cetim (technical center for mechanical engineers) and other players in the metal cutting sector. (Sandvik Coromant)

1.1.2 ISO 13399

ISO 13399 is an international technical standard for information exchange about cutting tools and tool holders between various software, as computer-aided design (CAD), computer-aided manufacturing (CAM), computer-aided engineering (CAE), tool management
software, product data management (PDM), manufacturing resource planning (MRP) or enterprise resource planning (ERP), and other computer-aided domains.

The development of ISO 13399 started in the mid 90’s. (Jurrens, 1995) This standard provides a unified format for cutting tool information, which can be used in any particular software or specific terminology. As long as tools are defined based on the standardized format, the information conversion quality and speed will be a significant boost, and the system will run smoothly. The final effect is that this standard improves the efficiency of the transform and the communication quality. At the same time, it also frees people from the manual work needed to be done before applying this standard. It makes everything easier for users.

However, as the emergence of multi-function tools, more and more tools intend to be used for more than one purpose. The initial version of ISO 13399 became too limited to support the complex tool information, which hardly fit in the existing implicit and inconsistent nomenclature. Therefore, the need for tool manufacturing company to find a way to communicate information about the use of a cutting tool and its properties seems a must. All these factors drive the need of the new version of ISO 13399. (Nyqvist, 2008) The new version of ISO 13399 makes achievable the complex communication between items and assemblies as well as multi-function tools.

In addition, with the new version of ISO 13399, the ability to describe new tools becomes possible. People can define tools with multiple functions, like a tool with one hand drilling cutters and the other hand milling cutter. Sometimes, the tools do not even exist, but the standard can describe them in advance. That shows the power of this ISO standard. (ISO 13399-1:2006(E), 2006)

Besides, ISO 13399 provides the base and condition for unifying communication language. It greatly improves the communication efficiency and reduces the communication cost. In addition, one language for communication (one mapping) is more reliable than multiple languages, inasmuch as it allows avoiding the deviation caused from the conversion process.

ISO 13399 helps offer a safe and convenient communication environment for: (Olsson)

- Tool item: tool item version, definition, property values, mutual relationship between each other etc.
- Tool assembly: tool assembly definition, properties, assembled instructions, extra information for 3D model software (CAM/CNC).
- References to external documents: 3D model information of single tool or tool assembly.
- Multi-function: tools being used for more than one purpose, multiply function tools.
- Nominal and physical tool: nominal tool information to CAM and tool room. Besides, physical tool information between tool room and CNC.
ISO 13399 aims to achieve providing a unified communication channel for suppliers, tool manufacturers, distributors and customers (Figure 1). From another perspective, it offers a mutual structure for cutting tool information exchange in diverse systems. ISO 13399 involves in different aspects as identification, machine tools and workpiece features, materials, machining operation and process, machine preparation, and product planning etc. (SIS-ISO/TS 13399-60, 2014)

Figure 1 — Relation among ISO 13399 and other user perspectives

1.1.3 Communication requirement

As it is mentioned before, communication between companies is always an issue since different software has different information model. Each software has its own standard of presenting and transforming data. In order to conduct communication, the system needs to understand the formats, or the rules of the other system. Therefore, for the possibility of communication from different systems, a standardized information format or model needs to be applied.

Product data management (PDM) is the business function often within product lifecycle management (PLM) that is responsible for the management and publication of product data. Today PLM is defined as: (Crow, 2016) (Ivica Crnkovic, 2003)
• A business method to offer solutions for cooperation innovation, regulation, publicity, and information definition.
• The development of the bridge between suppliers and customers.
• The management of the overall track and record cross the entire product development, from concept to the end of product life.
• A collective management of people, product, information, and process etc.

The cutting tool information talked about in this project also needs to follow the instruction in PLM, meets the requirements in the overall strategy. These drive the demand for ISO-10303 STEP.

1.1.4 Communication tool

ISO 10303 Industrial automation systems and integration — Product data representation and exchange is an ISO standard for creating a theory for the description of product manufacturing information representation and exchange. This international standard enables the whole life cycle product information being described in any particular system without attention of the format exchange.

Particularly STEP is a representative of the information transformation between computer-aided design, computer-aided manufacturing, computer-aided engineering, product data management/enterprise data modeling and other CAx systems. Data covered by STEP format involves in various fields from mechanical and chemical concept design, geometric representation, to production and processing, moreover, information about specific areas like construction, automotive, aerospace, marine, and non-renewable raw material related production etc.

The copyright of STEP is owned by the ISO, and it is not freely accessible. Therefore, in this case, the 10303 EXPRESS schemas are recommended to be used for implementation. The format of common STEP file is fully covered in ISO 10303-21 Clear Text Encoding of the Exchange Structure. (ISO 10303-21, 2002) ISO 10303-21 also gives a detailed instruction on how to conduct information based on a given EXPRESS schema.

STEP Physical File and p21-file are two interchangeable names of a STEP file. In this thesis application’s development process, the file extension .p21 has been used for common convenience. (ISO TC184/SC4 Secretary, 2003)

Besides the thesis application use, ISO 13399 also defines the presentation of product data information based on ISO 10303. This ISO theory is not only used in neutral format exchange, but defines the data library implementation and communication for ISO 13399 achievement. The related ISO 10303 files applied in ISO 13399 creation are stated as follows:
ISO 10303–11 provides the instruction for EXPRESS format.
ISO 10303–21 and ISO 10303–28 define the data exchange format
ISO 10303–40 to ISO 10303–56 describe the integrated resources

1.1.5 Communication benefit

As an alternative neutral format, STEP provides a solution for inconvenience of interaction between different software. Instead of the need of changing the format before communicating, STEP enables the users to represent and exchange information without considering format compatibility. (Brunnermeier, 1999)

STEP is superior to some other format translators by several contributions. Firstly, it can support a wide variety of products in different required aspects, as design, development, production, and the follow-up information etc. Secondly, instead of being a single-functional translator for two software systems, STEP provides a neutral model with universal and extended function to meet the needs. Finally, since STEP becomes an international standard, companies from different continents can communicate through this way. (Brunnermeier, 1999)

1.1.6 ISO ToolMaker

ISO ToolMaker application and its “function modules” are used for verifying and validating the development and implementation specification of ISO 13399. It provides an efficient way for creating and exporting assemblies to CAM or other simulation software. By using this software, the users can build their own library by selecting the tools they need for daily work and storing them in the interface. The tool files are used for assemblies’ creation, and the final assembly will be exported to CAM or other simulation software and viewed in 2D or 3D forms. The assembly files created by this application should follow the regulation stated in ISO 13399 and can be used in any computer-aided systems.

1.2 Purpose

The purpose of the thesis is to promote the reform of ISO ToolMaker, adjusting existing uncompleted functions, adding new modules for advanced usage, and reorganize the software structure for continuous improvement. Julie Huang, a product information specialist in Sandvik Coromant, has developed the first version of ISO ToolMaker application. The software framework has been basically set up. Since the software design is still in the
early stage, many aspects are not built in order and the way it should be, a systematic method is urgently needed to make an action plan and sort out this problem. The methodology applied in this project will be described in detail in the second chapter – Method.

1.3 Main features of the information model

Before discussing the main technique and strategy for building this ISO ToolMaker application, here are some must understand concepts about the main features of information models.

1.3.1 Different types of items

For the enormous numbers of cutting tools ordered and manufactured in the industry, at the initial developing stage of ISO 13399, the developers saw a ticklish problem about how to classify and manage this tool information. Thanks to concept of “modularization”, the tool manufacturers finally came up with an idea of breaking down a tool into different pieces. By utilizing the fact that the definition information of different type of items are reciprocally repulsive, this problem has been fully solved. (ISO 13399-1:2006(E), 2006)

Figure 2 shows the division of a normal cutting tool, the main groups.

Figure 2 — examples of the main groups (ISO 13399-1:2006(E), 2006)
These four types of items combined together represent the whole tool information between the workpiece and tool machine. The detailed classification is as follows: (ISO 13399-1:2006(E), 2006)

- Cutting item: It conserves the information of the cutting edge of a tool. Cutting edges mostly refer to inserts and the head of solid tools (e.g. solid carbide mill and carbide drills)
- Tool item: It contains the data of tool holders of tool items. A tool holder indicates a solid bar or solid head which is used for holding inserts or solid tools in place.
- Adaptive item: It holds the information for adapters, which is an adaptive connection between a cutter and a machine. The most widespread adapter for manufacturing is chuck.
- Assembly item: It represents the data of small joints. A joint is an important component to join different parts together and complete an assembly. The common used joints for tools are screw, collect, clamps, etc.

Different combination possibilities and the related Assemblies are shown in Figure 3. (ISO 13399-1:2006(E), 2006)

![Diagram of different types of items and assembly](image)

Figure 3— Different type of items and assembly (ISO 13399-1:2006(E), 2006)

The relation and possible connection among different types of items are shown in table 1 below: (ISO 13399-1:2006(E), 2006)
<table>
<thead>
<tr>
<th>Machine Side</th>
<th>Workpiece Side</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Cutting Item</td>
</tr>
<tr>
<td>Cutting Item</td>
<td></td>
</tr>
<tr>
<td>Tool Item</td>
<td>X</td>
</tr>
<tr>
<td>Adaptive Item</td>
<td>X</td>
</tr>
</tbody>
</table>

Table 1 — Relation and possible connection among different types of items

Assembly item does not show in the table since it is the joints used between every two connected parts.

1.3.2 Applied example

The four items shown in the blue boxes in Figure 4 have been used for ISO ToolMaker examination. Figure 3 and Table 1 indicate the connection possibility and related assembly. The four items are the input of the application, and the assembly 3 is the final outcome. The final assembly should be possible to compile in one single step and then should be able to be stored and exported as individual (as product data and geometry).
Figure 4 – Assembly structure example

The connection order follows the relationship rules shown in Table 1. CCMS (connection code machine side) and CCWS (connection code workpiece side) indicate the connection side of each item.
2. Method

This chapter describes two main methodologies that are chosen and used in the thesis project along with the work result. After finishing this chapter, readers should be able to understand the meaning of the methodologies, the reason behind the selection of each method, how it is being used in the project and its guiding effect to the project work.

Since this thesis project is a continuation and improvement of the previous work for the ISO ToolMaker application, so several execution steps have been applied such as problem solving procedures, old system updating and new function installation. There are some structured methodologies and scientific principles can be followed to contribute to an efficient work in a right direction.

2.1 Problem solving methodology

EBA Toolkit as an academic theory package is widely used for scientific and practical problem solving. The kit consists of techniques for different key areas (leadership, problem solving, economics, time management skills, and change) (EBA ) Within the problem solving category, there are several widespread branches:

- Six sigma tools
- 5 whys
- Fishbone diagram
- Total quality management (TQM) tools
- 5-step approach
- Failure mode and effects analysis (FMEA)

In this case, the 5-step approach is chosen to define and provide possible solutions for the application completion.

The 5-step problem solving approach helps with the problem identification and root causes tracing. It works as a simplified substitute for PDCA (plan-do-check-act/adjust), which provides users an efficient way to identify and resolve the problem.

The 5-step problem solving approach is shown as follows:
This problem solving methodology needs to be conducted step by step with the designed order. If the result does not work as expected, users need to move back to the second step and choose another practicable plan.

- **Problem identification:**
  Define the problem, be aware of the exact reasons, how it happens and what is the impact etc.
- **Potential plans:**
  Put out 2 to 3 feasible methods regarding to the problem identification.
- **Feasibility and result:**
  Evaluate each method and analyze the advantages, disadvantages and its possible outcomes.
- **Implementation:**
  Choose the best potential plan with the optimized results and execute the chosen plan in step 3.
- **Verification:**
  Inspect the final outcome and the intermediate process.

### 2.2 5-step approach application

The superior performance for a methodology is its general applicability. At the beginning,
5-step approach has come up in business domain, committed to help business people win over the quick problem solving competition. Since the ability and efficiency to solve a problem reflects the strength of a company, how quickly a company can understand, solve or eliminate a problem becomes the sore of this competition. With the development of this method, it is gradually used in some other fields like engineering problem solving, even daily life problem dealing. Its adaptability becomes wider and wider.

In this case, the problem arisen during the development of ISO ToolMaker can also fit in this 5-step problem solving approach.
### Problem identification

The existing ISO ToolMaker is not user friendly enough and does not provide desired functions. Some modules need to be modified and redesigned.

### Potential plans

1. Modification of the old interface based on the original code.
2. Development of our own interface.

### Feasibility and result

1. **Advantage:** No need to rebuild the code structure; The existing code can be used as a reference.
   
   **Disadvantage:** More efforts will be put in existing code learning; It will be extremely complicated, since most of the modules need to be modified and restructured.

2. **Advantage:** The new interface is fully under control; easy to modify in the future.
   
   **Disadvantage:** A new code structure is needed.

### Implementation

The second plan is chosen; Execution.

### Verification

Use CAM or other simulation software to view the results in 2D or 3D formats; Compare the output .p21 files with the previous application results.

Table 2 – Implementation of 5-step problem solving approach.

### 2.3 Optimization goals

Since the project aims to develop a more superior and user-friendly application which can benefit both the company and the users, as well as the future developers. Therefore, before the work is conducted, all the parties ought to be considered. In this case, TPS, Toyota
production system, which is known as Toyota’s “people and customer – oriented philosophy” is applied to set the optimization goals.

TPS is not a mandatory company procedure but an improvement concept/progress, which needs to be conducted during the work and have been validated during day by day. (Toyota) Continuous Improvement is the core of TPS.

Toyota puts forward three desires from three different angles: (Toyota)

- For customers: provide highest quality vehicles, at possible lowest price, with shortest possible lead times.
- For members: offer work satisfaction, job security and fair treatment.
- For company: ensure the flexibility to respond to market variation, develop cost reduction activities to achieve long-term profit.

As known, Qualitative change is the result of the accumulation of quantitative change. Based on these three targets, Toyota breakdowns the final goals into small pieces and puts effort in day-to-day work. In this case, for this project, three objectives have also been set to lead the work into the right track.

- For users: provide concise interface, sufficient info-text for process instruction, simplified operation.
- For developer: build logical code structure, facilitate the follow-up work.
- For company: ensure free access, power of use and revision.
3. Analysis and result

This chapter provides the results of the ISO ToolMaker application development, the description of main algorithms, interfaces and properties, as well as principles of assembly compilation. After reading this chapter, readers should be able to understand the results of this master’s thesis project, and the difference with the previous version of the ISO ToolMaker.

3.1 Assembly compilation

The principle of assembly compilation is based on the properties of every item. Every item has two sides: workpiece and machine sides, and has connection codes, which are matched to the codes of another item.

There are three possible connection codes:

- CCWS - connection code workpiece side.
- CCMS - connection code machine side.
- IIC - insert interface code.

The application logic compares these codes and defines the side to add the item. The CCWS code is broken into small part and compared to CCMS. The last four signs differ and represent side and tolerances, but the previous signs have to be the same to be matched. IIC codes matching logic follows the same principles, in addition, using the property of item type, and eliminate the addition to cutting or tool item on the workpiece side.

Adaptive items have more codes on workpiece side - to provide more alternatives to adapt the other item. Tool items and cutting items do not have the CCWS code as far as they are usually edge items.

According to the applied example, every item has its own CCWS and CCMS codes. One item can have several CCWS - the possibility to adapt to different items on the workpiece side, and one CCMS, which can adapt to another item.

Table 3 represents connection codes of the items.
<table>
<thead>
<tr>
<th>Item</th>
<th>CCWS</th>
<th>CCMS</th>
</tr>
</thead>
<tbody>
<tr>
<td>BIG-PLUS CAT-V to CoroChuck 930</td>
<td>ZYL18G2000****</td>
<td>SKG37C04000$$$</td>
</tr>
<tr>
<td></td>
<td>ZYL18R7874****</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZYL13G2000****</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZYL13R7874****</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZYL01G2000****</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZYL01R7874****</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SZC02G2000****</td>
<td></td>
</tr>
<tr>
<td></td>
<td>SZC02R7874****</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZYL10G2000****</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZYL10R7874****</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZYL20G2000****</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZYL20R7874****</td>
<td></td>
</tr>
<tr>
<td>Collet</td>
<td>ZYL01G1600****</td>
<td>ZYLO1G1600h06</td>
</tr>
<tr>
<td></td>
<td>ZYL01R6299****</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZYL10G1600****</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZYL10R6299****</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZYL13G1600****</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZYL13R6299****</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZYL18G1600****</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZYL18R6299****</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZYL20G1600****</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ZYL20R6299****</td>
<td></td>
</tr>
<tr>
<td>Cylindrical shank to CoroMill</td>
<td>SAT01A0120****</td>
<td>ZYL01G1600h06</td>
</tr>
<tr>
<td></td>
<td>SAT01B0120****</td>
<td></td>
</tr>
<tr>
<td>CoroMill 316 solid carbide head for</td>
<td></td>
<td>SAT01A0120$$$</td>
</tr>
<tr>
<td>square shoulder milling</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table 3 – Connection codes of items</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Table 4 shows the matched connection codes of</td>
<td></td>
<td></td>
</tr>
<tr>
<td>every item within the assembly.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Item</td>
<td>Type</td>
<td>CCWS</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>BIG-PLUS CAT-V to CoroChuck 930</td>
<td>Adaptive item, detail</td>
<td>Szc02G2000****</td>
</tr>
<tr>
<td>Collet</td>
<td>Adaptive item, detail</td>
<td>Zyl01G1600****</td>
</tr>
<tr>
<td>Cylindrical shank to CoroMill</td>
<td>Adaptive item, detail</td>
<td>Sat01A0120****</td>
</tr>
<tr>
<td>CoroMill 316 solid carbide head for square shoulder milling</td>
<td>Detail, tool item</td>
<td>Sat01A0120$$$$</td>
</tr>
</tbody>
</table>

Table 4 — Applied example assembly’s matching codes. Based on these codes, the assembly is compiled.

3.2 ToolMaker 1.0

The previous version of the ISO ToolMaker application has been created by Julie Huang in 2014. The application has the basic features needed for the cutting tool data representation and the function of assembly compilation – ToolMaker. The application consists of the main window and ToolMaker interface. Items and assemblies are imported to the main window. This interface shows all data of the item/assembly.

Figure 6 – The main window interface
The assembly is compiled in the ToolMaker interface, allows to export the file in two formats. It is possible to export as STEP-file format (ISO 10303-21), which represents 3D model of the items and assembly. Moreover, the export to .p21 file (ISO 13399), which keeps the assembly data and can be readable again in the application, is available.

The existing system of compilation allows creating the needed file, but takes many steps, which have to be done manually by user.

The algorithm is:

1. Select an item on the machine side. Every item has a few lines which correspond to the different CCWS (connection code workpiece side) codes. The user has to go through these lines manually, looking for the item, which can match this item.

   ![Figure 7 – ToolMaker 1.0 interface, selection of an item on machine side](image)

2. Select an item on the workpiece side. In our case, the items only have one CCMS (connection code machine side), which can be connected with CCWS codes.
Figure 8 – ToolMaker 1.0 interface, selection of an item on workpiece side

3. Compile the subassembly, and export it as ISO-13399. The user has to save this intermediate file, and import it to the application again. All these steps are also done manually. The user has to know exactly where the file is saved.

Figure 9 – ToolMaker 1.0, an export of the intermediate assembly file

4. Repeat steps 1-3, but already with the imported sub-assembly on one of the sides.
Figure 10 - ToolMaker 1.0, a selection of item on machine side, and assembly on workpiece side

In such a way, concentration, solid knowledge of items and assembly, and time are required for the creation even one file. From the user point of view, this application is not friendly, some steps have to be done manually permanently, and there is an obvious lack of automation.

To sum up, there are the main problems of the application:

- No keeping track of item and assembly files.
- The need of file search for import every time.
- Time-consuming creation of assembly, manually, step-by-step.
- The lack of information: new user has to spend a lot of time for understanding the mechanism and principles of work.

3.3 ToolMaker 2.0

The new interface has been created taking into account the previous version logic algorithms and saving mechanisms, however, this interface is more user-friendly and understandable.
3.3.1 Algorithm

The compilation algorithm has been developed based on the analysis of all possible situations, which can occur during the assembly compilation.

Items are taken in a bulk, and every item is checked for the side positioning and matching with another item. Items can be compiled with each other, or to the existed assembly. In a normal flow of the process, item can be added on machine or on workpiece side. This operation is provided automatically. In certain cases, an item can fit to the both sides, thus, the confirmation dialog occurs.

When one item is added, the next item is taken, and then it goes through the same processes. If an item cannot fit to any side, the warning occurs, and the application is terminated. When all items are added, the new assembly is compiled, and can be exported.

Figure 11 represents the algorithm of the assembly compilation graphically.
3.3.2 ToolMaker 2.0 interface

The window of the interface is divided on two parts: Items and Assembly. Imported items are added to Items. This part includes two columns: Name and Ordering code. Assembly includes the following columns: Item, Ordering Code, CCWS (connection code workpiece side), Dependency, CCMS (connection code machine side), IIC (insert interface code).
It is possible to create one assembly at a time. When at least one item is added to the assembly, the tree of available items is redrawn, and only items, which can be added are left. When all possible items are added, there are no items.

The text-info informs the user about an item possible position. It helps to understand how the application works and what happens by clicking on the Item. There are three cases how an item can be added:

2. Workpiece side.

Figure 14 – ToolMaker 2.0, the addition of item on the machine side

<table>
<thead>
<tr>
<th>Name</th>
<th>Ordering Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cylindrical shank to CoroMill®</td>
<td>E12-A16-SS-065</td>
</tr>
<tr>
<td>BIG-PLUS CAT-V to CoroChuck™ 930-VB40-HD-20-082</td>
<td></td>
</tr>
</tbody>
</table>

BIG-PLUS CAT-V to CoroChuck™ 930 can be added on the machine side
3. Both sides. In this case, the dialog window appears, which let a user to choose the side where the item should be added. Another assembly example is provided.
The main benefit of the new application is that it becomes possible to add items on both sides without manually saving of intermediate files. In addition, the application logic checks matching codes and all possible positions and inform the user. In such a way, a user always knows how the final assembly will look like.

There is a function of multi-selection of items. It is possible to add all items by one click, if the items are matched and can create the assembly. This function allows to reduce time of assembly composition significantly.

Figure 16 – ToolMaker 2.0, Confirmation dialog for the side selection

Figure 17 – ToolMaker 2.0, One-click assembly
3.3.3 Renaming of the assembly

There is an added feature of renaming of sub-assemblies during the assembly compilation. It can simplify the following search in My Assemblies interface, and represents needed data (for example, name, the target use, machine, etc.).

However, it is possible to rename assemblies when the assembly is compiled using single items. When the assembly is saved, the mechanism of saving does not allow renaming the existing assembly or its parts. In such a way, to fix it and make possible to change names of existing and saved sub-assemblies, it is needed to change the whole mechanism of saving the file in .p21 format and the structure of data used in the application. In such a way, the problem is very deep in the roots, therefore, it is possible to use this function only during the
compilation, before saving and export of the file.

3.3.4 Export of the assembly

The export of created file is carried out by the same saving mechanism as previous version of the application. It was very important to get the same result at the end, because these files would be used by another application after, and, in such a way, they have to follow the standards and schema.

Another benefit of the new version of the application is that the exported as ISO 13399 format .p21 assembly is directly imported to the main window. It can be added to My Assemblies interface, without additional search and import of the files.

![Cutting tool data](image)

**Figure 20 – Demonstration of auto-import to the main window after ToolMaker 2.0**

3.3.5 Editing of the assembly

By the right-click on the item/assembly in Main Window, it is possible to select Edit in ToolMaker 2.0.
The ToolMaker 2.0 interface would be opened, and the selected object would be added to Assembly part directly. The application allows to add on the machine side new items to the assembly, if available.

It is also possible to change names of unsaved sub-assemblies.
To sum up, there are obvious benefits and significant increase of convenience during the using of the application. Many manually steps and processes are automatized and proceed without user participation. However, the user is always informed, there is a good communication between human and computer.

3.4 My Items and My Assemblies interfaces

My Items and My Assemblies interfaces have been developed based on the decision to make ISO ToolMaker more user-friendly. The previous version of the application obliges to import files from the original folders, each time the application is opened. This procedure takes a lot of time, as far as a user has to find the file, which is supposed to use for the assembly compilation. Another problem is that the user cannot see the properties of the file, only the file name. The properties is possible to check only when the file has been already imported. It can be a reason of the following mistake, or taking additional time for searching a necessary file.

The new version of the application represents two new interfaces: My Items and My Assemblies. First, the main reason is a need to keep all item and assembly data in one place. The principle of these interfaces is based on the idea of having catalog of the available items and assemblies. Secondly, the interfaces make easier to select different items and assembly from different folders in the file system, significantly reduce the time of searching the item or assembly in the file system, and keep track of total amount of items and assemblies in your library.

Figure 23 – ToolMaker 2.0, editing of assembly, demonstration of renaming feature
How to add Item/Assembly to My Items/My Assemblies interfaces:

1. Import files from the folder by clicking on "File" -> "Import".

Figure 24 – ToolMaker, Import files

2. Click on "Add" button, the file browser is opened. Select files.

Figure 25 – ToolMaker, File browser

3. Click on "OK" button. Files are imported. It is possible to import items and assemblies at once.
4. The addition of an item/assembly is carried out by the right click on the line in the main window. Click on "Add to My Items"/"Add to My Assemblies".

<table>
<thead>
<tr>
<th>Item Name</th>
<th>Item Type</th>
<th>Ordering Code</th>
<th>Instance Co...</th>
<th>Orgi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cylindrical shank to CoroMill® EH adapt.</td>
<td>adaptive item, detail</td>
<td>E12-A16-SS-065</td>
<td>1</td>
<td>Sanc</td>
</tr>
<tr>
<td>Collet (eng)</td>
<td>adaptive item, detail</td>
<td>393.CGS-20 16 52</td>
<td>1</td>
<td>Sanc</td>
</tr>
<tr>
<td>CoroMill® 316 solid carbide head for sq.</td>
<td>detail, tool item</td>
<td>316-12SM450C12010P 1030</td>
<td>1</td>
<td>Sanc</td>
</tr>
<tr>
<td>BIG-PLUS CAT-V to CoroChuck™ 930 (eng)</td>
<td>adaptive item, detail</td>
<td>930-VB40-HD-20-082</td>
<td>1</td>
<td>Sanc</td>
</tr>
<tr>
<td>Assembly 1 (eng)</td>
<td>assembly, tool item</td>
<td>930-VB40-HD-20-082+393.CGS-20 16 52+... 1</td>
<td>(rol)</td>
<td></td>
</tr>
</tbody>
</table>

### Figure 27 – ToolMaker, Menu of adding to My Items/ My Assemblies interface

My Items interface is represented as a table, and includes following columns:

1. Item Name
2. Item Type
3. Ordering Code (unique code)
4. Organization
5. In File (file path)

Figure 28 – My Items interface

The data is taken from the Item Properties. The choice of these features is based on the idea that a user can find the necessary item easier and faster. The keeping data can help to recognize the needed item, for example, by ordering code, which is unique, or by its name. In addition, it is possible to sort items by columns, for example, by Item Type, to simplify the search.

Figure 29 – My Items interface, the sorting function
My Assemblies interface is represented as a table with following columns:

1. Item Name. This column represents the name of assembly, sub-assemblies, and items.
2. Item Type
3. Ordering Code
4. CCWS (connection code workpiece side)
5. Dependency
6. CCMS (connection code machine side)
7. IIC (insert interface code)
8. In File (file path)

Every assembly is possible to unfold, and check the properties. Columns 4 – 7 show the information about assembly matching codes and dependencies between items.

![My Assemblies interface](image)

Figure 30– My Assemblies interface

Added features:

- It is possible to carry out a multi-selection, and select a few or all items.
- The application can recognize an item and an assembly. In case of multi-selection, by clicking on "Add to My Items", only items would be added to the relevant interface, and vice-versa.
- If an item/assembly is already added to My Items/My Assemblies, nothing happens. It is defined by file path, to exclude double-adding.
- The windows and columns are resizable; it is possible to change the view according to user needs.
- There is a counter of items/assemblies added to the relevant interface, to simplify the track of all files in the catalog.
- After compiling and saving the assembly in ToolMaker2.0, the assembly is directly imported to the Main Window and can be added to the My Assemblies interface.
- It is possible to open or copy URLs from Documentation table in the main window interface by right-click on the line.

The file is imported to the main window by using file path to the file. By adding to "My Items"/"My Assemblies", the file path is recorded, and can be used again. File data is recorded to the external file in comma-separated values (.CSV) format. For example, the information about item in My Items can look like: 000000003d5fbbd264c13290fbbf8cc8a,,"Cylindrical shank to CoroMill® EH adaptor (eng) "","adaptive item, detail",E12-A16-5S-065,Sandvik Coromant (role: id owner),,,C:\Users\Zenbook\Desktop\testDoc\from IMTS demo\5732629.p21. There are two auto-generated files for each interface. The files are auto-saved and available at the next launch of the application.

![Image](http://example.com/image.png)

Figure 31 – My Items interface after next launch of the application

The decision to choose this format has been based on the following factors:

- The .CSV format allows to record the data from the table, each line has its own value, and commas separate values within it.
- The file is human-readable, and can be opened separately.
- The accessibility of the format on different platforms is another driven factor. Many
external applications can accept this format as well.

- Because of the format simplicity, it is easy to debug the file and program in case of problems.

How to import file from "My Items"/"My Assemblies" interfaces:

1. Select an item/assembly in a relevant interface. It is possible to select a few or all items/assemblies.
2. Click on "Load and Close" button. The files will be imported, and the interface window is closed. The user is informed if the file does not exist or is not available.

![Image of My Items interface]

**Figure 32 – My Items interface, demonstration of files import**

There is "Remove" button in the interfaces. The functionality allows removing an item/assembly from the interface's .CSV file. The record of the file information can be deleted; the file itself is not deleted.

To sum up, My Items and My Assemblies interfaces help to keep all file paths to the files in one place, without browsing the whole file system for searching the needed items and assemblies. The file management significantly reduces the time of assembly compilation and the application becomes more user-friendly.
4. Further work and development

This chapter provides the ideas for further development and potential properties to add. The ideas can be used for another master’s thesis project, for example. After reading this chapter, readers should be able to understand and imagine all the potential of ISO ToolMaker.

4.1 3D models file management

At this moment, the assembly 3D model in .STP file format can be saved in case of presence of relevant .STP files for every item. The folder with these documents has to be kept in the application folder. In such a way, a user has to copy existing .STP item file to the certain folder on his computer manually. This step can be time-consuming and confusing. The user has to know exactly where this folder is; in addition, there is no file management, the user has to know also what file is needed for what, for example, before deleting.

However, it is possible to download .STP files from the internet to the needed folder. At this moment, some of the item files contain the information about 3D model and the direct link to this file. However, there are still a lot of files, which do not represent this information.

In such a way, there is a need of new interface or functions, but its realization can be a long-term project, as far as not every item is represented by a link to the file. For example, it could be a function, which downloads 3D models of imported as temporary files, which have the same behavior as system temporary files.

4.2 Additional export

Probably, the need of additional formats of export can be realized. For example, after tool assembly, it is possible to add export function as Excel file. It could be implemented for better communication between production, supplier and customer. Bill of Materials (B.O.M) can be exported for the following product lifecycle management, etc. using the available and appropriate format.

4.3 Adveon™

The existent application Adveon™ is a commercial application, which is used by Sandvik Coromant. The application represents tool library according to ISO 13399, and allows compiling the assembly for the further using, for example, for CAM or other simulation purposes, etc.

The basic functions and features of Adveon™ have been used as example for Tool Maker 2.0. However, ISO ToolMaker is an open-source application, and can be used by other companies, which strive to ISO 13399. In such a way, by this application, many global goals can be reached. The expansion of ISO 13399 use leads to easier communication between all
the production parties, and, as a consequence, to the quality increase.

4.4 Tweeting machine

With the rapid change of the whole industry and the emerge of the concept of Industry 4.0 (smart factory), mechanical industry dedicates to find an automation and intelligent data exchange in manufacturing technologies. Nowadays Internet of Things gets the enormous boost and perspectives of use.

It is possible to add more communication properties to ISO ToolMaker application. New features can be represented as tweets for receiving and sending data and information.

Receive tweet:

- Machining operation. Tools differ for every operation, and machining operation can be the first criteria for tool assembly.
- Design criteria and tolerances. The choice of the tool properties has to take into account the geometry of the operation and a workpiece.
- Constrains. The tool has to be fit into the machine, and executes all supposed operations.

Send tweet:

- Optimized tool. The automated assembly of the tool has to provide the optimized tool as a response tweet.

The development of ISO ToolMaker application can lead to new extended horizons within industry and cutting tool data communication.
5. Bibliography


