TECHNISCHE UNIVERSITÄT MÜNCHEN
Extraordinariat für Anwendungen der virtuellen Produktentwicklung

Generative Design-to-Fabrication Automation using Spatial Grammars and Heuristic Search

Christoph Heinrich Walter Ertelt

Vollständiger Abdruck der von der Fakultät für Maschinenwesen
der Technischen Universität München
zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs

genehmigten Dissertation.

Vorsitzende: Univ.-Prof. Dr.-Ing. Birgit Vogel-Heuser

Prüfer der Dissertation:
1. Univ.-Prof. Kristina Shea, Ph.D.
2. Prof. Chris McMahon,
   University of Bath / UK

Die Dissertation wurde am 28.02.2012 bei der Technischen Universität München
eingereicht und durch die Fakultät für Maschinenwesen
am 13.07.2012 angenommen.

ISBN 978-3-00-040177-0

Acknowledgements

The following publications are part of the work presented in this thesis:

Ertelt & Shea 2008

Ertelt, C.; Shea, K.: Generative design and CNC fabrication using shape grammars, Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE. Brooklyn, New York, USA, August 3-6 2008.

Ertelt & Shea 2009

Ertelt, C.; Shea, K.: An application of shape grammars to planning for CNC machining, Proceedings of the ASME 2009 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE. San Diego, California, USA, August 30-September 2 2009.

Shea et al. 2008

Shea, K.; Engelhard, M.; Ertelt, C.; Hoisl, F.: A Framework for a Cognitive Design-to-Fabrication System, Proceedings of the 7th International Symposium on Tools and Methods of Competitive Engineering (TMCE 2008). Ismir, Turkey, 21. – 25. April 2008.

Shea et al. 2010

Shea, K.; Ertelt, C.; Gmeiner, T.; Ameri, F.: Design-to-fabrication automation for the cognitive machine shop. Advanced Engineering Informatics 24 (2010) 3, p. 251-268.

Ertelt et al. 2009

Ertelt, C.; Rühr, T.; Pangercic, D.; Shea, K.; Beetz, M.: Integration of Perception, Global Planning and Local Planning in the Manufacturing Domain. In: Grau, A. et al. (Eds.): 14th IEEE International Conference on Emerging Technologies and Factory Automation, Mallorca, Spain, September 22-26. ISBN: 978-1-4244-2728-4.

Bannat et al. 2011

Bannat, A.; Bautze, T.; Beetz, M.; Blume, J.; Diepold, K.; Ertelt, C.; Geiger, F.; Gmeiner, T.; Gyger, T.; Knoll, A.; Lau, C.; Lenz, C.; Ostgathe, M.; Reinhart, G.; Roesel, W.; Ruehr, T.; Schuboe, A.; Shea, K.; Stork genannt Wersborg, I.; Stork, S.; Tekouo, W.; Wallhoff, F.; Wiesbeck, M.; Zaeh, M. F.: Artificial Cognition in Production Systems. Automation Science and Engineering, IEEE Transactions on 8 (2011) 1, p. 148-174.

ABSTRACT

Generative Design-to-Fabrication Automation using Spatial Grammars and Heuristic Search

Planning for Computerized Numerical Control (CNC) fabrication requires generation of process plans for the fabrication of parts that can be executed on CNC enabled machine tools. To create such plans, a large amount of domain specific knowledge is required to map the desired geometry of a part to a manufacturing process, thus decomposing design information into a set of feasible machining operations. Approaches to automate this planning process still rely heavily on human capabilities, such as planning and reasoning about geometry in relation to machining capabilities. In this thesis, the author presents a new, spatial grammar-based approach for automatically creating fabrication plans for CNC machining from a given part geometry. To avoid the use of static feature sets and their pre-defined mappings to machining operations, the method encodes knowledge of fundamental machine capabilities. The use of spatial grammars as a formalism enables systematic formulation of hard and soft constraints on spatial relations between the volume to be removed and the removal volume shape for a machining operation. Further, a software implementation of the core method is presented and validated using several examples of machining a part on a milling machine including changed tools and tool failures during runtime. Advanced heuristic search methods are investigated to further improve the quality of the generated plans in terms of accuracy and machining time. Overall, the approach and method presented is an enabler for the creation of an autonomous fabrication system and CNC machine tools that are able to reason about part geometry in relation to available capabilities and carry out on-line planning for CNC fabrication.

Generative Konstruktions- und Fertigungs-Automation mittels Räumlicher Grammatiken und heuristischer Suchmethoden

Die Planung von CNC-Bearbeitungsprozessen erfordert die Erzeugung von Prozess-Plänen die auf CNC Werkzeugmaschinen ausgeführt werden können. Eine große Menge spezifischen Wissens ist erforderlich um die Form eines Bauteils den entsprechenden Fertigungsprozessen zuzuordnen und damit die Gestalt-Information in eine Menge möglicher Fertigungsoperationen zu zerlegen. Existierende Ansätze sind zu großen Teilen noch abhängig von menschlichen Fähigkeiten, wie Planen und Schlussfolgern über die Geometrie im Bezug zu den Fertigungsmöglichkeiten. In dieser Arbeit wird ein neuer, auf räumlichen Grammatiken basierender Ansatz vorgestellt um automatisiert Pläne für die Herstellung einer Bauteil-Geometrie auf CNC Werkzeugmaschinen zu erzeugen. Die Methode beinhaltet das Wissen über die fundamentalen Fertigungsmöglichkeiten, um statische Zuordnungen zwischen Features und Fertigungsoperationen zu vermeiden. Die Verwendung von räumlichen Grammatiken erlaubt es, Randbedingungen und Beschränkungen bezüglich der Form eines Bauteils und den möglichen Fertigungsoperationen systematisch zu formulieren. Weiterhin wird ein Software-Prototyp der Kern-Methode vorgestellt und anhand mehrerer Szenarien validiert. Weiterentwickelte heuristische Suchmethoden werden untersucht um die erzielten Ergebnisse hinsichtlich Genauigkeit und Herstelldauer zu verbessern. Insgesamt ermöglicht der Ansatz, dass autonome Fertigungssysteme und CNC Werkzeugmaschinen zwischen der Bauteilgeometrie und den verfügbaren Bearbeitungsmöglichkeiten abwägen und selbstständig, während der Laufzeit, Pläne für die CNC Bearbeitung generieren.

Table of Contents

1 Introduction

In order for companies to remain competitive in today's global market they must be able to rapidly respond to changing customer needs and desires. Manufacturing companies are constantly increasing the number of product variants they offer to the market to react to the growing trend to produce customized products (LINDEMANN et al. 2006). The customization of products drives the need for manufacturing systems to more efficiently produce parts with non-pre-defined geometry at a competitive price. Even today, the process, from design to the actual fabrication of parts, relies on human experts for planning, preparing and carrying out fabrication processes. This process is presented in Section 1.1, focusing on the part fabrication itself using CNC machine tools. For each new part to be fabricated, new machine tool programs must be created that drive the machines tools. These part programs are required for every part and shape to be machined and often are prepared manually or created using software tools operated by human experts. The role of human experts in today's design-to-fabrication process chain is presented in Section 1.2. In the context of customized and individual part production, today's conventional automation is too inflexible and often leads to a significant drop of productivity due to frequent changeovers. To overcome this, researchers propose to apply cognition within technical systems that allows the automation systems to increase the productivity despite the increase in complexity (PUTZER & ONKEN 2003) by becoming more autonomous. This can be achieved by providing the cognitive system with an internal representation of its environment and giving it the capability to reason about the knowledge within this representation. The new paradigm for enabling autonomous behavior of systems in dynamic environments, called "Cognition for Technical Systems (CoTeSys)", is presented in Section 1.3. The drivers and barriers for fully automating the design-to-fabrication process are presented in Section 1.4. This chapter closes with a presentation of the thesis outline in Section 1.5 and a summary of the targeted research contributions in Section 1.6.

1.1 The design-to-fabrication process

Process planning refers to the selection of the procedures required to convert a part design into a piece of hardware (MACHINABILITY DATA CENTER 1986). Throughout the whole process chain, shown in Figure 1-1, from an initial design to the finished part, decisions have to be made and plans created as well as machine instructions generated. The first step is the creation of a geometric model of the design using Computer-Aided Design (CAD). Computer-Aided Process Planning (CAPP), as the essential link between CAD and Computer-Aided Manufacturing (CAM) (CORNEY et al. 2005), is a tool for preparing the fabrication by generating high level process plans. It involves recognition of possible machining operations from the geometric model, the selection of machines and processes as well as the determination of their sequence. Overall, DRAGHICI & BONDREA (1998) consider CAPP as comprising preparation, design, and coordination of manufacturing.

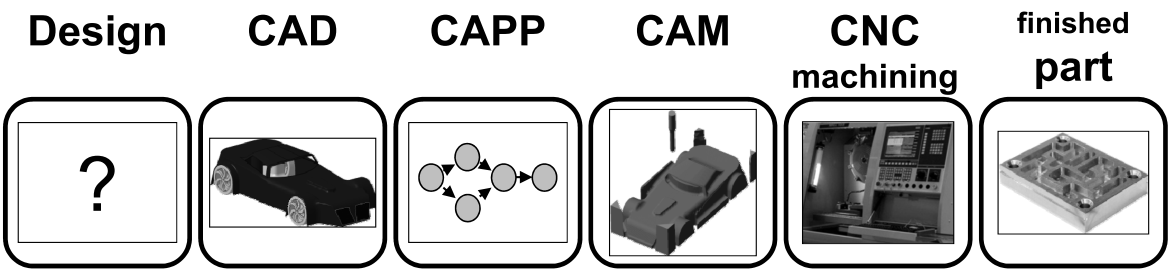


Figure 1-1: Current Design-to-Fabrication process (CORNEY et al. 2005, FEENEY & FRECHETTE 2002, KRISHNA & RAO 2006, MIAO et al. 2002, SHEA et al. 2008).

Different approaches to CAPP can be identified (CHEN et al. 2006): The variant approach relies on using process plans from previously fabricated parts within the same part family and to allow the human expert to change the process plan as to fit it to the part to be fabricated. The similarity between several parts is determined based on a classification scheme for parts such as Group Technology (GT) (KUSIAK 1990). However, this approach is tedious, labor-intensive and must rely on the similarity and classification of the fabricated parts. In contrast to this, the generative approach creates new process plans for each part using knowledge about the fabrication processes.

In contrast to CAD/CAM, CAPP focuses more on the production oriented aspects than the geometric reasoning involved in fabrication planning. However, sometimes researchers use the term CAP/CAPP and CAD/CAM alike. For the remainder of this thesis, the terms are used independently either for managing process plans in the case of CAPP or for the link between geometry and operations and processes in the case of CAD/CAM.

Table 1-1: Sample process plan (according to ZHANG & ALTING 1994)

<table>
<tr><td colspan="2">Part No. S0125-F
Part Name Housing</td><td colspan="6">Date 08.06.2007
Material Aluminium, 82 N/mm²
Stock L 30.00mm x W 30.00mm x H 20.00mm</td></tr>
<tr><th>No.</th><th>Process description</th><th>Machine</th><th>Setup</th><th>Tool</th><th>Parameters (Speed, Feed)</th><th>NC-Data</th><th>Time [min]</th></tr>
<tr><td>10</td><td>Mill bottom surface 1
Roughing</td><td>MILL01</td><td>see attached #1
for illustration</td><td>Face mill #1</td><td>5200 Rpm
7925 mm/min</td><td>Housing_Bottom</td><td>3 setup
5 machining</td></tr>
<tr><td>20</td><td>Mill top surface
Roughing</td><td>MILL01</td><td>see attached #1</td><td>Face mill #1</td><td>5200 Rpm
7925 mm/min</td><td>Housing_Top</td><td>2 setup
6 machining</td></tr>
<tr><td>30</td><td>Drill 4 holes
Roughing</td><td>DRL02</td><td>set on surface 1
see attached #2</td><td>Fwist drill #5</td><td>3900 Rpm
364 mm/min</td><td>Housing_Holes</td><td>2 setup
3 machining</td></tr>
<tr><td colspan="8">(...)</td></tr>
</table>

To enable the fabrication of a part in an industrial environment, a process plan for its fabrication must be derived from the part design, reflecting the available resources and machines. An example process plan can be found in Table 1-1. The process plan can be provided digitally or on paper. The header section of the process plan contains the part identification number, the part name, the date of the creation of the process plan, a specification of the workpiece and the size of the stock or workpiece. Each row of the table corresponds to a fabrication operation, each specified by a sequence of operations, a description of the process, the machine tool, the setup of the part within the machine tool envelope including fixtures, the tool used, cutting conditions (speed and feed), a reference to the NC program and the time duration of setup and machining (CHEN et al. 2006, ZHANG & ALTING 1994).

For fabrication planning of parts, several important tasks have to be fulfilled. In the following a selection of the most important tasks within the design-to-fabrication process is presented.

Checking the machinability

Before a process plan for machining a part can be created it must be ensured that it can be machined with the available resources. To assess the machinability three factors have to be considered (MIAO et al. 2002): *Part size*, *Complexity* and *Accessibility*.

Part size determines whether the part can fit inside the machine tools and fixtures and whether the machine tools and handling devices can support the dimensions and weights of the part. The *Complexity* of the part is greatly determined by the types of features, i.e. geometric entities e.g. prismatic pockets, freeform surfaces, used (see Section 2.1 for a more thorough definition of the term feature). An additional determination is whether they can be machined only from one access direction or from multiple. The more potential access direction a feature has the less complex the part is since the number of setups can potentially be reduced. For *Accessibility* it is important that every feature can be reached collision free by the tools of the machine tool such that the machined part matches the designed part.

However with increasing part complexity, the checking of the machinability relies on software tools to analyze the outcome of the process. That means the process plan is created iteratively until the desired result, a feasible process plan, is achieved.

Setup planning

Setup planning deals with the definition of the location and orientation of the part within the machine tool such that machining can take place (YAO et al. 2007). The machining features, and respectively their machining operations with the same approach direction are grouped into the same setup (AMAITIK & KILIÇ 2007) to minimize change-over time on the shop floor. Therefore alternative feature interpretations reducing the number of setups are chosen (SHAH & MÄNTYLÄ 1995).

Selection of machine tools and operations

The operations required to machine features need to be specified since the same feature can be machined using different types of operations. An example for this is a hole that can be milled using circular interpolation, bored using a boring tool or be drilled. Further the machine tools need to be defined. Considering a similar example, a concentric hole in a cylindrical part can be drilled on a turning machine or on a milling machine. If the type of machine tool is determined, e.g. "milling machine", the actual machine tool (i.e. its instance) to use on the shop floor has to also be defined, e.g. "milling machine 1."

Specification of manufacturing sequence

After the type of operations and machine tools have been defined, the sequence of operations can be specified aiming at a minimum machining time (MIAO et al. 2002). The operations can be reordered depending on operation constraints and precedence constraints. Operation constraints dictate, that operations are carried out in a distinctive order, e.g. finish machining after rough machining or reaming after drilling. The precedence constraints reflect the spatial arrangements of the operation, i.e. can an operation be carried out collision-free before another (MIAO et al. 2002). In literature, the term cutting sequence (SAKAL & CHOW 1994) is used similarly to the term machining sequence.

Selection of cutting tools

The selection of cutting tools comprises the determination of the cutting tool type and tool size regarding the part's geometry and machining operations such as drilling, turning or milling. The tool selection greatly influences process time and part quality (MIAO et al. 2002). The tool selection defines also technological parameters such as the tool material, e.g. carbide or high speed steel (HSS) (SAKAL & CHOW 1994).

From the plan created, the machine instructions for the Computerized Numerical Controlled (CNC) machine tools can be generated by first generating tool paths and then translating these to controller specific instructions using Computer-Aided Manufacturing. These instructions can then be executed on the CNC machine tool to fabricate the desired part.

The origin of CAM stems from the introduction of Numerical Control (NC) (BESANT & LUI 1986) and later Computerized Numerical Control (CNC) machine tools. With increasing capabilities, such as multi-axis machining, programming and validation of programs became more and more difficult. To overcome this, the first CAM tools were developed offering a more convenient way of programming using a graphical user-interface and program validation using simulation. According to KOCHAN (1986), CAM is the "computer-aided preparation of manufacturing including decision-making, process and operational planning… and manufacturing with different types of automation" and further encompasses the management and control of the manufacturing system (XU & HE 2004). This also involves the technical control and operation of machines and tools (based on NC and CNC technology) during production and not the organizational control as in Production Planning Systems (PPS) (FRANZ 1991). Giving a single concise definition of CAM is difficult due to the widespread and sometimes ambiguous use of the term CAM (ZHANG & ALTING 1994). However, CAM is used now as general term for any computer tool to support the creation of machining programs (SHIRASE & FUJII 2009). Therefore the terms CAM and CAD/CAM are used synonymously if not indicated otherwise.

The idea of coupling Computer-Aided Design (CAD) Systems with Computer-Aided Manufacturing (CAM) lies in using the existing product definition data for process planning and NC programming (GRABOWSKI & ANDERL 1990). The integration of these subsystems on a high-level leads to CAD/CAM (HATVANY 1982). Therefore CAD/CAM overcomes the need for traditional manual or computer-assisted NC program creation even when higher level languages such as APT are used.

In a CAD/CAM application, the part program can be created with graphic feedback, able to display the part geometry and the created tool paths. After the verification of the tool path it can be converted to the program for the CNC machine tool (CHANG et al. 1991).

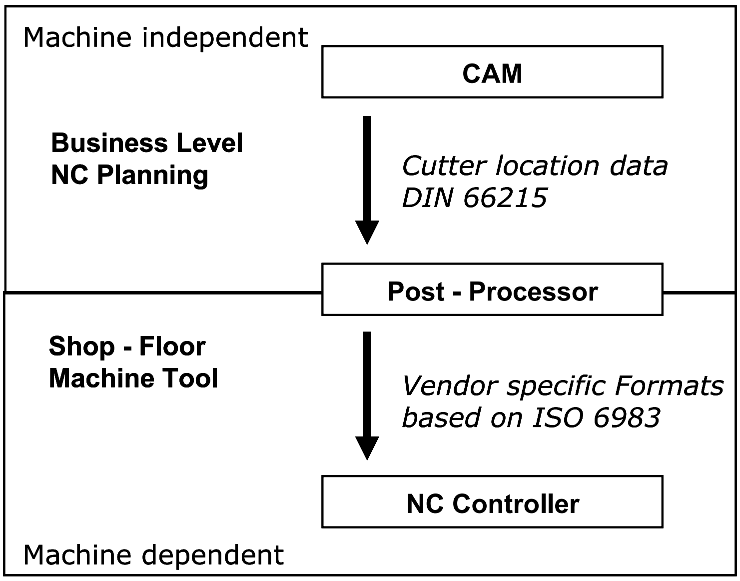


Figure 1-2: State-of-the-art NC Programming (according to WECK & WOLF 2003)

The current state-of-the-art in NC programming is shown in Figure 1-2. At the Business Level NC Planning, CAM tools are used for programming. The result is machine independent Cutter Location data (CL data) and standardized in DIN 66215 (1974). CL data describes the toolpaths in terms of locations, type of operations, feeds and speeds. This data, however, cannot be used directly by machine tools. Therefore, it is transformed into the vendor specific dialect of G-code using a specialized post-processor.

Despite the efforts to develop more powerful languages to program machine tools such as Binary Cutter Location (BCL) (ALBERT 1990) or STEP-NC (ISO 14649 2003), G-code is still the primary language in today's manufacturing industry.

The working steps in a CAD/CAM application are as follows (according to GRABOWSKI & ANDERL 1990, MCMAHON & BROWNE 1993):

Resource definition

The geometry of the used workpiece and the tool is defined, potentially selected from a library. Also, the machine tool is specified in terms of number of axis and type of machining process (e.g. turning or milling).

Recognition of manufacturing features

Feature recognition (FR) can be generally defined as processing of a geometric model from a CAD system to find portions of the model matching the characteristics of interest for a given application (SHAH et al. 2001). Manufacturing feature recognition therefore searches the geometric model for manufacturing features, i.e. geometries that correspond to machining operations and that can be removed from a stock to yield the desired part. The goal is to build a manufacturing feature model of the part to be fabricated which can then be mapped to available machining operations and resources. A detailed discussion of automated feature recognition can be found in Section 2.1.

Machining operation sequencing and tool path generation

After defining the resources, the sequence of machining operations is identified and tool paths are created interactively by the human expert by specifying the part contours to be machined in each machining operation. To machine the desired part, the tool of course has to be guided correctly along the desired surface to be machined (ZEID 1991). The correct trajectory is called toolpath. In traditional CAD/CAM, the toolpath can be generated from the specification of the drive surface (guiding the tool), the part surface (the intentionally created surface) and the check surface (motion constraint). The toolpath can be generated if the part region to be machined is specified and the cutting strategy, e.g. zig-zag, spiral in/out, is defined. Generally a minimum length of the cut is desired for minimum machining time (CHANG et al. 1991).

Definition of Approach and retract directions

The approach of a machining region of the workpiece and the retract direction of the tool must be defined to avoid collisions of the tool with the workpiece. This also involves defining the retract and re-approach direction during the transitioning from one machining region to the other. This decision is usually carried out by a human expert planner (MIAO et al. 2002).

Generation of Cutter Location (CL) data

After the full definition of the desired tool movements, a Cutter Location (CL) data file is created by the CAD/CAM software. This file describes the location of the cutting tool in relation to the specified coordinate system.

Simulation of machining process

The verification of the created CNC program plays an important role in the planning process. The toolpaths have to be verified regarding the desired output in terms of the created shape and the program checked for possible collisions during execution (ZEID 1991).

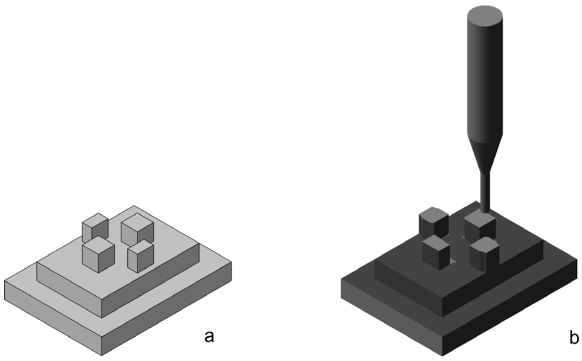


Figure 1-3: Designed part (a) and simulated outcome of the CNC program (b)

Figure 1-3 shows a part as it was designed (a). With this design, the CNC-code was created using a CAD/CAM software. The result has been simulated as shown (b). However, the shape of the part obtained from the simulation significantly differs from the originally designed part, especially regarding the four blocks on the topmost surface. In practical CAM, every program therefore has to be checked by a human expert and interpreted to ensure its correctness.

Post-processing of CL data to machine specific code

To allow the execution of the NC program on the machine tool, it must be created by post-processing the CL data file. Through this, the location and movements of the cutting tool are translated into movement commands for the individual axes of the machine tool.

Adding further NC data

After the NC program generation, it can be necessary, due to deficiencies in the post-processor used, to add further, machine tool specific, commands to the NC code file.

The most common CNC programming language G-code, created by Gerber Scientific Instruments, a manufacturer of photoplotters, was approved as DIN 66025 in 1983 and ISO 6983 in 1988. It contains preparatory (G), miscellaneous (M) and control (F Feed, S Speed and T Tool) functions that are combined into records. Each record corresponds to either a line on a punch card as in the first NC machines, or today, to a line of a text file. Each record starts with the letter N and an incrementing record number. In each record, one or several functions can be called with given numerical values. Optional comments can be added with a preceding semicolon.

Figure 1-4 shows an example of a CNC program in G-code. The program begins with selecting the reference coordinate system *G54* that was set to the upper left front edge of the workpiece in the machine tool controller. Next, a tool change is performed and afterwards the spindle is started counter-clockwise and the working feed is set to 100mm/min. After this, the virtual tool tip is moved to the home position above the part at rapid feed using the *G0* command. Then, a sequence of cutting operations, using the *G1* command for linear interpolation, is carried out. These operations create the toolpath depicted as dashed line. Before ending the program, the tool is retracted from the part and the spindle is stopped.

```
N10 ; select reference coordinate system
N20 G54
N30 ; Change tool to tool no. 6 and active edge 1
N40 T6 D1 M6
N50 ; turn on main spindle with 5000 rpm ccw
N60 S5000 M3
N70 ; set working feed to 100mm/min
N80 F100
N90 ; go to home position at rapid feed
N100 G0   X0.0  Y0.0  Z10.0
N110 ; perform cutting operation
N120 G1   X0.0  Y0.0  Z-5.0
N130 G1   X20.0Y0.0  Z-5.0
N140 G1   X20.0Y20.0Z-5.0
N150 G1   X0.0  Y20.0Z-5.0
N160 G1   X0.0  Y0.0  Z-5.0
N170 ; retract tool
N180 G0   X0.0  Y0.0  Z10.0
N190 ; stop spindle
N200 M5
N210 ; end program
N220 M30
```

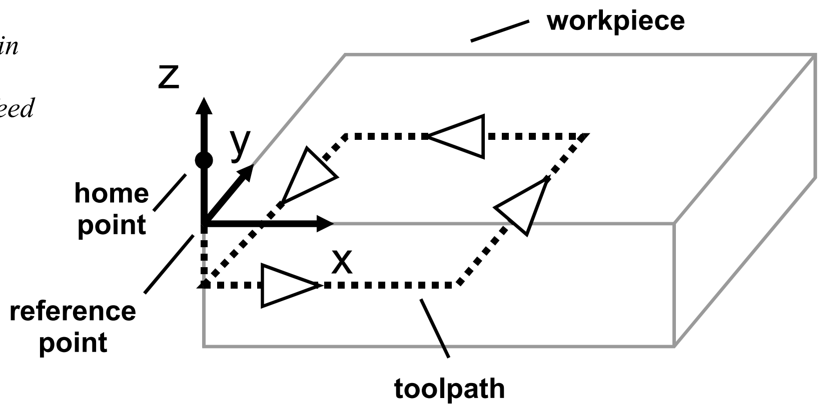


Figure 1-4: Example of a G-code CNC program and created toolpath.

In the following the detailed working step that have to be accomplished during the use of Computerized Numerical Control (CNC) for fabrication are presented:

Selection and preparation of workpiece/manufacturing part

For the machining of the desired part, the workpiece must be provided. Depending on the type of manufacturing environment, this can also require the machinist to choose a raw part or take an existing raw part and machine it such that it matches the raw part shape as it was used during the creation of the NC program.

Tool setting

To machine the part correctly, the correct tools, as they were defined during the NC program creation, must be installed and, if required, calibrated such that the machine tool controller can compensate for tool diameter and tool length.

Selection of machining conditions

Often, the exact machining conditions for the part are determined on the shop-floor level (MACHINABILITY DATA CENTER 1986). This is usually done by looking up appropriate parameters in tables and fine tuning them with respect to surface quality, machine vibrations and sound. However, with more recent computer tools, the first estimation of machining conditions is transferred to the CAM planning stage.

Machine setup, part fixturing

The raw part must be fixed in the machining envelope according to the setup specifications in the process plan. To hold the part securely in place, the part is clamped to the machine table or held by a vise. For most methods of fixturing, the coordinate-system origin as it was used during programming must be accurately measured on the raw part and entered into the machine tool controller.

Simulation of machining instructions on the machine tool

Before the NC program is executed for the first time on the machine tool controller, the machinist checks the NC program to ensure its correctness and to prevent damage to the machine tool due to collisions or wrongly specified cutting conditions. This can be done by displaying graphically the tool movements, by graphical simulation of the NC program on the machine tool or by having a “dry-run” of the NC program without a part inside the machine envelope. The “dry-run” functionality runs the NC program without driving the spindle and having the axes operate at rapid feed all the time.

In the remainder of the thesis the most important steps for the fabrication plan creation are pursued further. These are: The selection of machine tools, specification of manufacturing sequence, the selection of cutting tools and the machining operation sequencing and toolpath generation. The checking of the machinability can also be considered by proving that a feasible process plan for part fabrication exists by generating it.

1.2 The role of human experts in design-to-fabrication

Despite the numerous efforts to automate the fabrication of parts and components, human experts still play an important role in the design-to-fabrication process. To illustrate this, a typical planning process and a study of a CAD/CAM/CAPP system is presented. This section finishes with a conclusion on the role of human experts in design-to-fabrication.

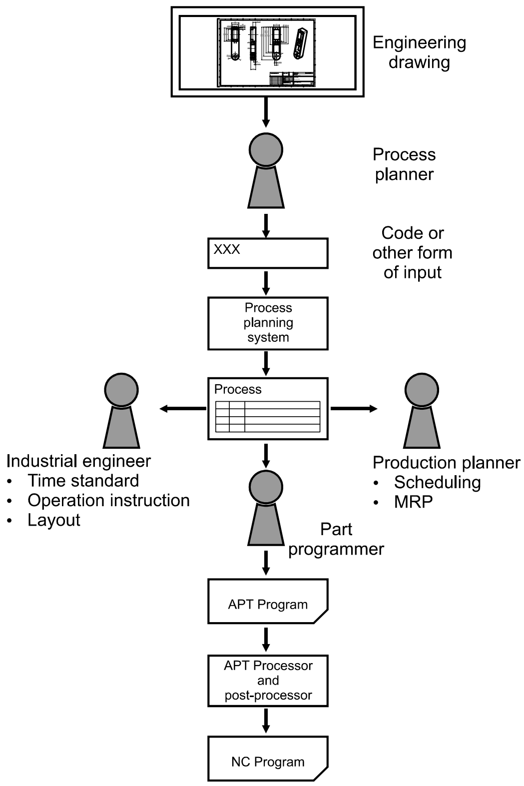


Figure 1-5: Typical process planning system by 1985 (based on Chang & Wysk 1985)

Figure 1-5 shows the course of process plan creation for a typical process planning system in the year 1985. The process begins with an engineering drawing of the part that should be fabricated. A process planner takes the information of the engineering drawing and uses some form of coding scheme to classify the part. Such a very simple classification could be whether it is a part for milling, a part for turning or a sheet metal part. Once the classification has been created, a standard plan for fabrication can be retrieved from the process planning system. This process or process plan, however, cannot be executed as it is still incomplete. An industrial engineer has first to set the time standard for fabricating the part, write the operation instructions and plan, if necessary, the machine or plant layout. After this the production planner can create the schedule for fabrication and carry out the Manufacturing Resource Planning (MRP).

Once the resources for manufacturing have been determined, the part programmer can begin to code a specific APT (Automatically Programmed Tool) program that can be used to fabricate the part on the CNC machine tools. This program is still in a high-level language and must be therefore post-processed to yield the machine tool specific CNC program that can be executed on the machine tool's controller.

To examine the role of humans in today's design-to-fabrication chain, a study on the use of two selected CAD/CAM/CAPP tools was conducted, namely CATIA CAD/CAM and Tebis CAD/CAM. This study is not intended to serve as a quantified study but intends to show the general level of existing automation as well as some of the barriers in the design-to-fabrication automation. Due to the length of the process presentation and similar result, only the results for Tebis CAD/CAM are presented.

The analysis of the Tebis CAD/CAM process was carried out, observing (video recording) and interviewing an experienced user while he was using the software to create a process plan for a 2.5 D milled part. The part is a prototype locator part for inserts in a carbon fiber composite.

The first step of the process, shown in Figure 1-6, is loading the part geometry from a CAD file and importing it into the software. This step is carried out by the human expert, while the conversion process itself is done by the computer automatically. After the surface model has been loaded, it must be checked for correctness by the expert, especially that it does not contain any "leaks", i.e. that it is watertight, or overlaps. Then the expert defines the origin of the setup that must match later the machining origin of the machine tool.

The machine tool that should be used for the fabrication is either defined or selected from a database of existing machine tools. Similar to this, the tool is defined or selected respectively. Again, the tool must match the actual tooling of the machine tool. Once the machine tool and the tool are defined, the machining process can be specified. This comprises the type of operation such as prismatic milling, turning or freeform surface milling.

Then, the stock for the part is defined regarding its geometry and the single machining operations are defined (e.g. face milling, pocket milling). The cutting conditions for each operation are specified either from the expert's knowledge, tables or technical specifications of the workpiece supplier or empirical formulas. For the complete definition, the cutting strategy must be defined. Examples are line-by-line, zig-zag or inside-out/outside-in strategies. To successfully apply these strategies, the machining area (the envelope in which the material removal action should take place) and the start- and end-plane are defined by the operator. If some islands of the part exist in the machining area, these have to be recognized and specified by the human expert. Up to this step, almost every operation has to be carried out with or solely by the human expert with aid of the software application.

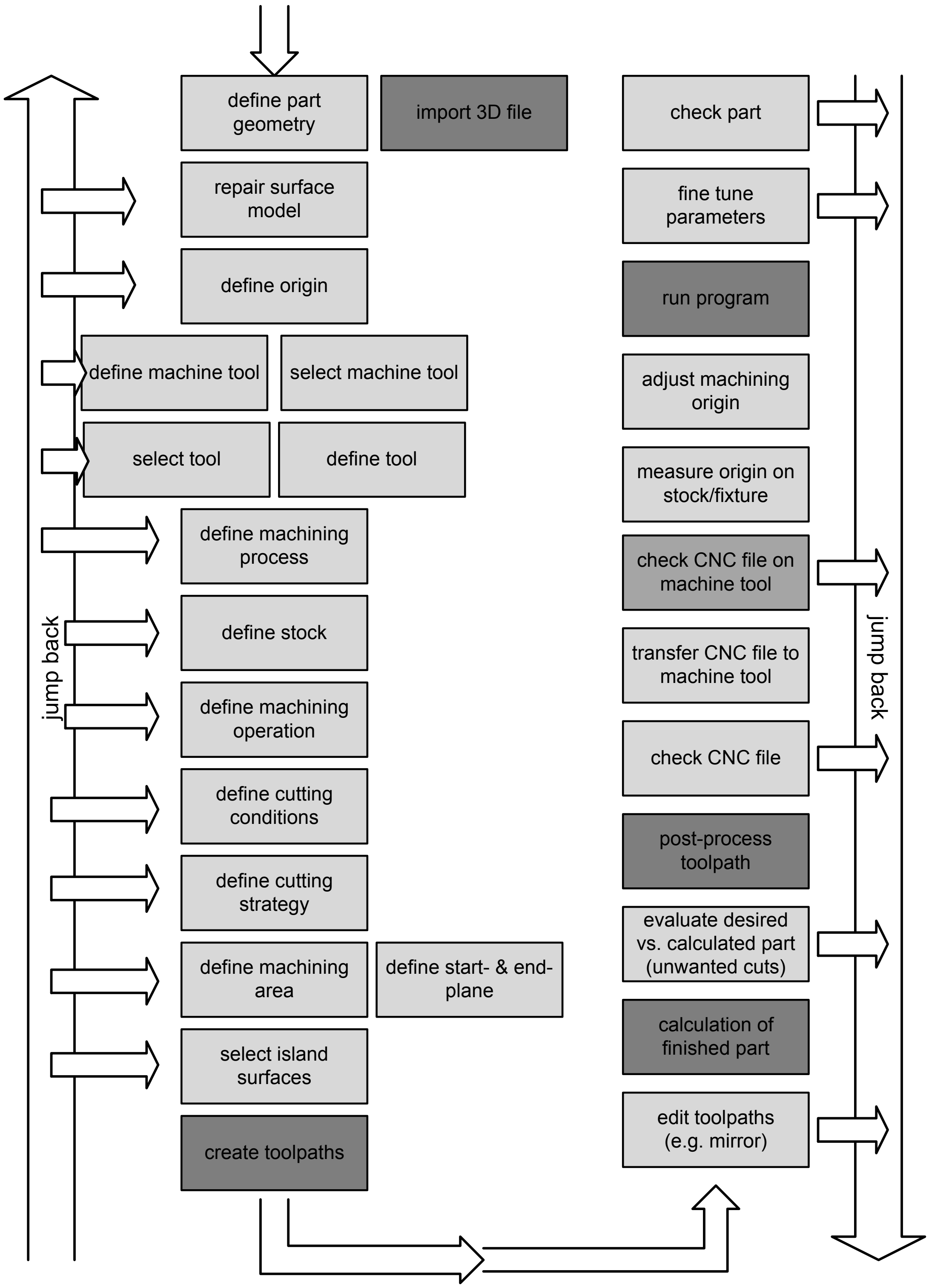


Figure 1-6: Study of process planning using Tebis CAD/CAM. Different shades of color indicate the degree of automation from human based (light gray) to machine based (dark gray).

The toolpaths can then be generated by the software and are presented to the human expert. He then edits the toolpaths, e.g. by mirroring, arranging according to a pattern, to avoid multiple definition of similar part areas. If the toolpaths do not meet expectations, the process must return to a previous step. After editing the toolpaths, the software can calculate the theoretical shape of the finished part based on the generated and edited toolpaths. The human expert evaluates the results and checks it for unwanted cuts. If the part's shape seems to be incorrect, the process returns to a previous step. If the expert found the part to be correct, the toolpaths are post-processed, i.e. they are converted into the CNC controller specific representation. The CNC code is then checked for correctness by the expert before it is transferred to the CNC machine tool where it is again checked for correctness by the machine tool controller and the machine tool operator. If the CNC code is not correct, changes must be made in previous steps. To hold the part during machining, a suitable fixture must be installed and its origin must be measured. The machining origin must then be adjusted accordingly. After the part has been inserted and secured in the fixture, the program is run by the machine tool. During the machining, the operator is able to fine tune the cutting conditions by changing speed and feed. After the program execution has terminated, the operator roughly checks the part for correctness. If the part has been machined correctly, the process is finished, otherwise it must return to previous steps of the process.

Though many claim that the design-to-fabrication process has been completely automated successfully, today's CAD/CAM tools still require a high level of human expert interaction to produce feasible process plans, just considering the use of a typical CAD/CAM system. Typically, the human process-planning expert is still required to determine setups, select fixtures and tools and sequence the necessary operations whereas the CAM system is mainly used to generate toolpaths (CORNEY et al. 2005).

The currently established design-to-fabrication process in industry still relies on expert knowledge and skills (CORNEY et al. 2005, SAKAL & CHOW 1994), as Figure 1-7 illustrates. Here, human expert knowledge and skills are required from design to Computer-Aided Design (CAD), Computer-Aided Process Planning (CAPP), Computer-Aided Manufacturing (CAM), Computerized Numerical Control (CNC) machining to the finished part. This fact is also supported by the concept of Computer-Integrated Manufacturing (CIM), which integrates all manufacturing activities by means of linked computer aids and data interfaces (MCMAHON & BROWNE 1993).

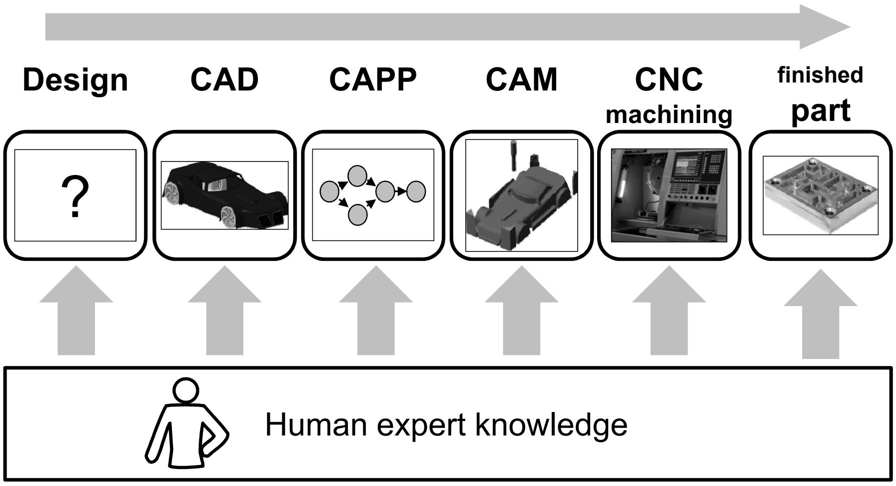


Figure 1-7: Current Design-to-Fabrication process (CORNEY et al. 2005, FEENEY & FRECHETTE 2002, KRISHNA & RAO 2006, MIAO et al. 2002, SHEA et al. 2008)

1.3 Cognition for Technical Systems (CoTeSys)

Cognition for technical systems is one approach that could make machines more autonomous, in a way that they can reason about and plan their own actions. Through this, the steps required in design-to-fabrication that are carried out by human experts, could be transferred to machines.

The notion of Cognitive Technical Systems goes back to the cognitivist paradigm as it was formulated by NEWELL (1994): Such intelligent systems, according to the unified theory of cognition (UTC), are based on symbolic representations following a set of syntactic rules. The grounding of these symbols is done by the programmer who formulated the symbols and rules; therefore it captures their particular knowledge of a specific domain and makes it accessible to the system (KEMPF et al. 2009). Therefore, the system itself does not have an understanding of the domain it is acting in.

Traditionally, in cognitivist systems, the act of cognition is regarded as being an identifiable and explicit process within an intelligent system (VERNON et al. 2007), as shown in Figure 1-8. The perception system describes the world using a computable model. The cognition then uses the model to somehow perform the “thinking” by symbol processing and perform actions based on the decisions made. This model corresponds to the top-down approach in AI. BROOKS (1999) explains the difficulty of this model: The cognition box can be recursively decomposed, finding in each recursion some intelligent behavior until only a simplistic computational intelligence is left which uses highly abstract facts hardly allowing performing truly intelligent actions. Once again, the system relies on a computable formalism and is therefore purely syntactic which denies any kind of semantics and therefore mind.

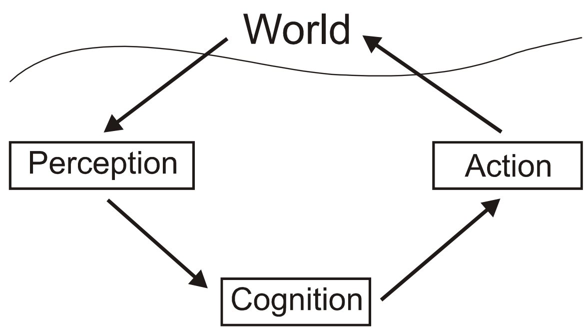


Figure 1-8: Traditional Model of Cognition (BROOKS 1999)

However, the so called new AI, also known as emergent systems approach (VERNON et al. 2007), moved away from this view of intelligent behavior. Cognition is regarded as a property of an intelligent system which emerges from the overlapping of perception and action and which is only implied by an observer (BROOKS 1999). The idea of the new AI is not to create a thinking machine that has a mind but tries to build systems that can react intelligently to their environments. This new AI basically has to suffer from the same problem that no understanding and mind can emerge from a formal system which is only syntactical as today's computers. The idea is that cognition might appear for an outside observer by the overlapping processes of perception and action, as shown in Figure 1-9. Therefore, cognition as the intelligent behavior of a technical system exists only implicitly or in the eyes of an observer, i.e. the system is not intelligent though it appears to act intelligently. This view corresponds to the behavioristic, bottom-up view on the intelligence of a system.

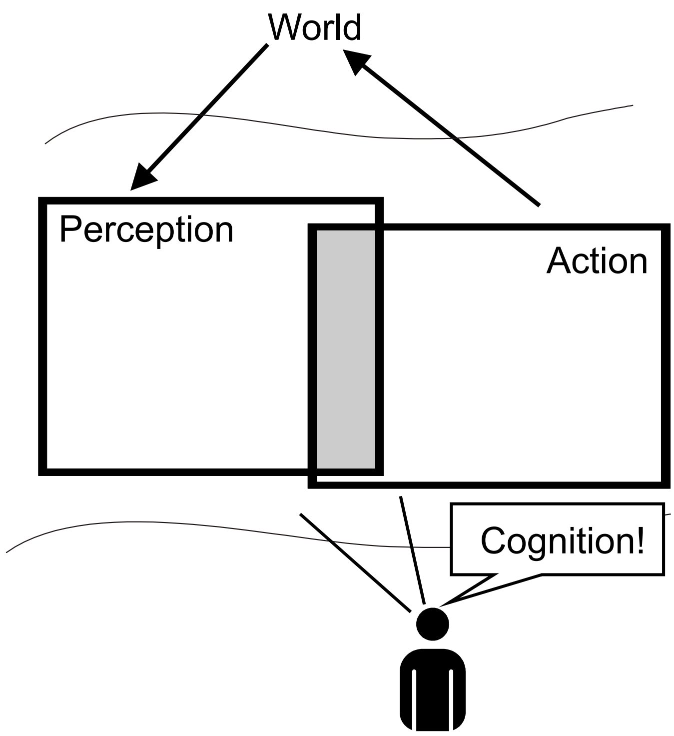


Figure 1-9: Model of cognition, where the cognition is in the eye of the observer within the overlap of perception and action (BROOKS 1999).

The commonality of all approaches to realize cognitive processes within technical systems are the areas of perception and action. The interplay of different areas of human cognition and their interaction is of most interest in cognition for technical systems. Through combining these skills, technical systems should be enabled to perform more autonomously, flexibly and robustly in a constantly changing environment without necessarily being able to exhibit intelligence. These technical systems should aid humans by solving problems that are difficult, cumbersome or even dangerous for humans.

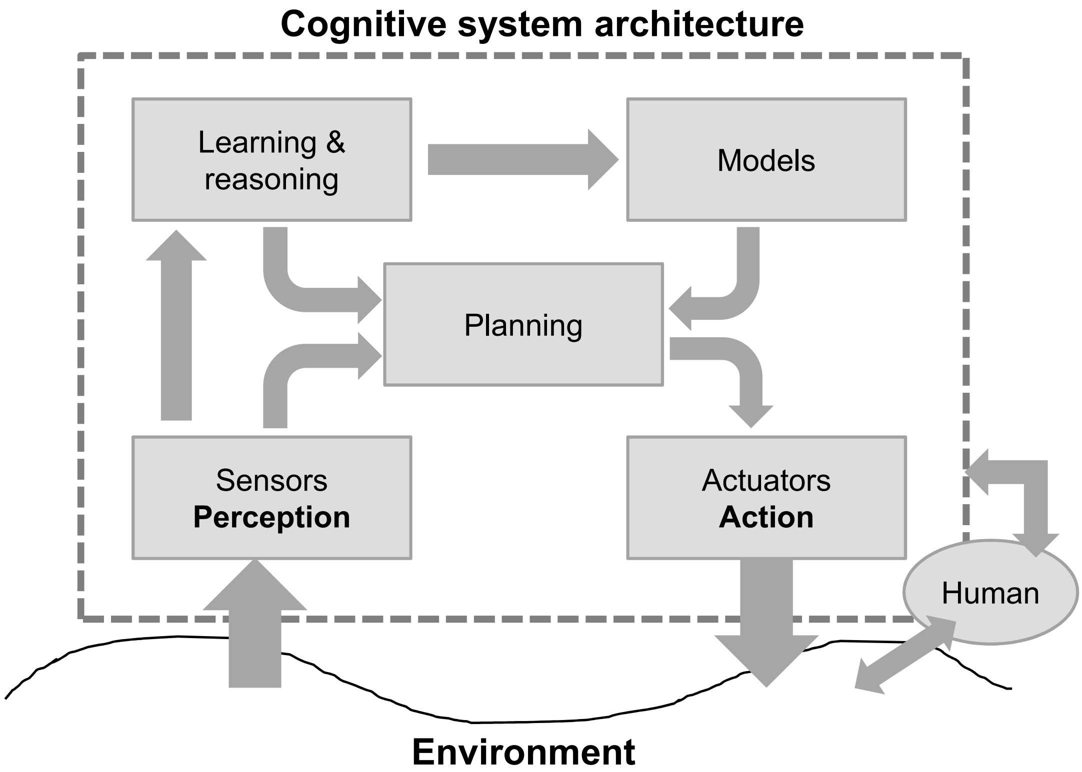


Figure 1-10: The perception-action closed loop (BEETZ et al. 2007)

The Research Cluster of Excellence "Cognition for Technical Systems – CoTeSys" (COTESYS 2011) addresses this matter by following the approach to cognition as depicted in Figure 1-10. A Cognitive Technical System, indicated by the dashed border, is able to read sensory data from the environment. The perceptual mechanism allows the system to map the sensor data to the internal representation of the Cognitive System. Learning enables the system to reorganize data contained in the internal representations or models beyond the existing explicitly formulated models. Reasoning capabilities allow the system to infer conclusions from a range of data. Models are one core aspect of cognitive systems since they are used by all other areas of cognition. Using models, that is partial representations of the environment or the system itself, future action effects and retrospective findings can be evaluated. This is particularly important for planning. Here, the future effect of actions that are directed to achieve a specified goal must be evaluated in order to find the right combination and sequence of actions to finally obtain the goal. After the plan has been created by the system, it can be brought into action using the systems actuators. A cognitive system should not only interact with its environment using its sensors and performing actions but also should interact with humans. A human in this context can be a user or instructor that should specify goals for the cognitive system or supervise it to allow the system to learn from its own performance.

Having these thoughts in mind it is still important to clarify the claim of this research: The aim is not to create a real thinking machine that has a mind but rather build a technical system that is able to autonomously and robustly perform under changing conditions of its environment and changing demands from the users it is interacting with.

1.4 Drivers and barriers for automated design-to-fabrication

At a base level design and fabrication is about producing physical objects or artifacts. However, due to the increasing complexity of parts and their fabrication within global manufacturing networks, producing a design becomes more and more complex as well. This factor and of course the ongoing trend towards individualized products drives the need for manufacturing systems to more efficiently produce parts and products with both customized geometry and customized configurations without the need for time-consuming, manual re-planning and re-programming. While this goal is supported in part by rapid prototyping and rapid manufacturing (see Section 2.3 for a discussion), a significant gap still exists in rapidly fabricating end-use metal parts.

The desire to increase the flexibility in mass production has led to the development of Flexible Manufacturing Systems (FMS) during the 1980s. These are capable of producing a family of parts, i.e. a set of parts with similar shape, size and manufacturing processes. However, the set of parts and the required processes are pre-programmed into the different hardware of the system. Since FMS are used for mid-volume and mid-variety fabrication of parts, the production of individual, one of a kind parts requires the time-consuming re-programming of the whole system. This does not only comprise changing the programming of the machine tools but also changing the programs of robots and handling devices to grip and position the parts correctly.

On the other hand, typical workshops run by humans can produce a large variety of different parts. The specialists in the workshop do not require an exact specification of the part to be produced. Often a sketch or a simple drawing can be enough. If a part cannot be fabricated, they can successfully find solutions by changing the design while maintaining the desired functionality. The specialists working in the machine shop learn and gain experience over time and use it to improve the quality of the parts produced. However, such machine shops are often slow and costly compared to more automated fabrication systems.

What if the cognitive capabilities of human specialist in a mechanical workshop could be transferred to an automated system such as a network of CNC machine tools? The system would be self-aware, knowing its own fabrication capabilities and autonomously plan the process to fabricate a new part based on the current state of the environment. Each machine would be enabled to learn over time, acquire new capabilities and improve its performance considering time and quality. If difficulties or even failures occur, the machines would re-plan from the current situation to overcome the difficulties.

Figure 1-11 Cognitive Machine Shop (CogMaSh) hardware setup.

Within the Research Cluster of Excellence Cognition for Technical Systems (CoTeSys), the Cognitive Machine Shop (CogMaSh), as shown in Figure 1-11, is under development. It is a cognitive system of machines such as CNC machine tools, conveyor belts and storage facilities as well as robots for handling and assembly. By having the machines planning and working together to produce a part, the high throughput of automated systems is combined with the robustness and flexibility of mechanical workshops run by humans. This goal is addressed in the long-term by integrating on-line knowledge models, perception, automated planning and learning. With this, the system is aware of its own fabrication capabilities and resources that it can use to autonomously plan the process to most efficiently make a new, previously unknown, part based on the current state of the environment (SHEA et al. 2010).

In its final development stage, CogMaSh should be able to produce new and individual parts upon request, without the need for re-programming the system. The machines will plan the fabrication together, requesting additional services such as a required workpiece or fixture from each other. If the system is confronted with difficulties it will attempt to overcome these. If for example, a machining capability is missing, a tool is broken or a workpiece is missing, the system will find an alternative plan enabling the successful fabrication of the requested part. Through shifting the planning process down to the machine level, planning and performing becomes an integrated process, thus allowing for dynamically generating and changing process plans on-the-fly according to recognized changes in the environment.

In a first step, knowledge models that represent machine capabilities, especially those of machine tools, need to be developed. These representations must be used to allow the machines to plan their own actions and perform these in their environment. Giving the machines the capability to react to changes in their environment is crucial. This first step is the focus of this thesis.

1.5 Thesis structure

The structure of this thesis is shown in Figure 1-12. After the introduction to design-to-fabrication in Section 1, Section 2 presents a review of the state-of-the-art in methods related to design-to-fabrication. Feature technology as a mediator between design and fabrication is reviewed along with existing approaches to intelligent systems for planning and manufacturing. In addition to this, Rapid Prototyping (RP) technology as an enabler for the convenient fabrication of customized parts is reviewed as a competing technology to autonomous design-to-fabrication. Spatial grammars with their applications to generative design in engineering are presented as well.

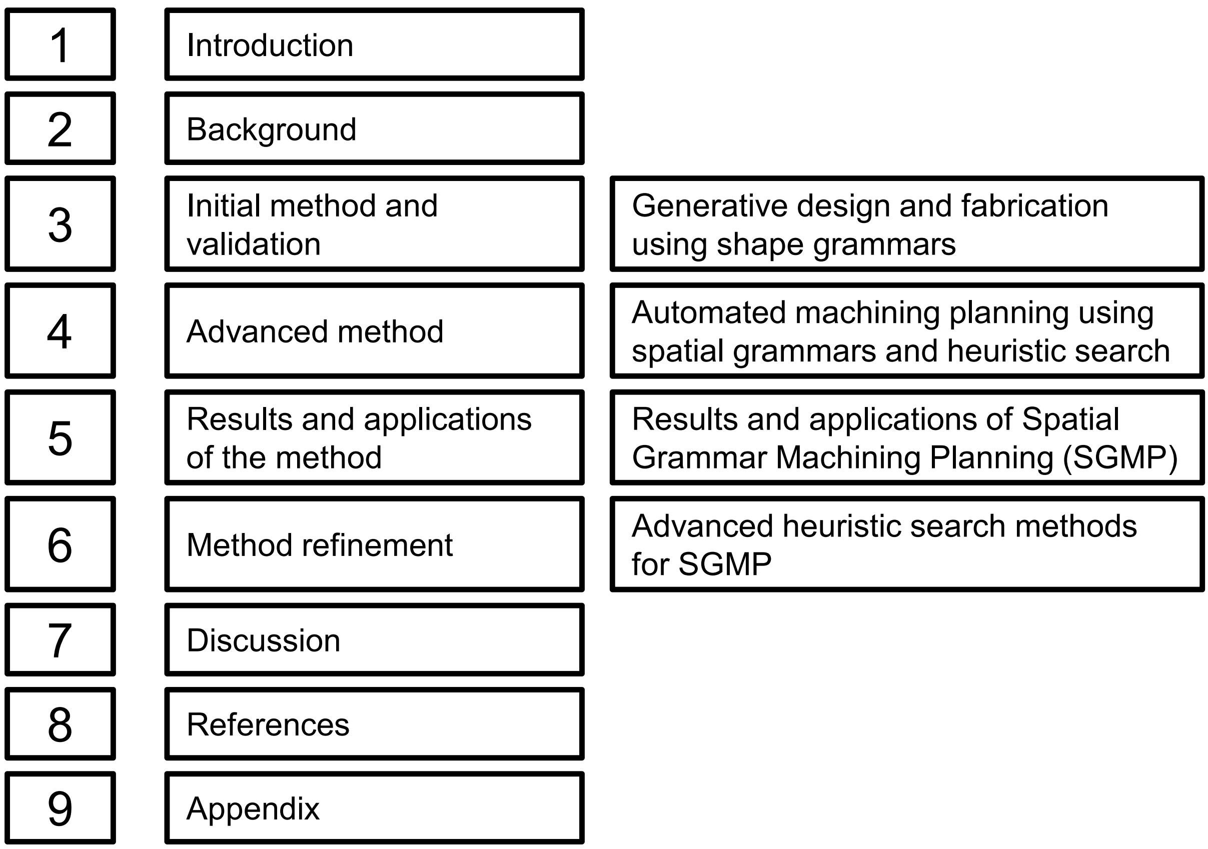


Figure 1-12: Thesis structure.

An initial method and validation for generative design and fabrication using shape grammars, is presented in Section 3. Here, designs are created by means of a shape grammar and then mapped to the manufacturing domain using a combination of different shape grammars. The machining knowledge is embedded within the shape grammar framework. An example part is presented to validate the approach and to prove its feasibility.

Section 4 presents a more advanced method to autonomous design-to-fabrication using spatial grammars and heuristic search methods to decompose CAD designs into feasible CNC machining plans. Within the decomposition process, a two level search process is applied to select rules of a spatial grammar and to find the best transformation of the applied shape. An initial implementation of the method is presented.

The results of the method along with a variety of scenarios that can be addressed by the method are presented in Section 5. These comprise the autonomous generation of machining plans, creating of machining plans for more complex parts, using multiple tools for the fabrication of a part, reacting to the unavailability of workpieces and reacting to tool failures.

Based on the previous presentation of the basic method, Section 6 presents more refined search methods, attempting to further improve the obtained results.

Finally, this work is concluded by reviewing the research contributions, discussing the limitations of the method and its implementation as well as giving recommendation for future work in the direction of this research in Section 7.

As appendices, a brief documentation of the implemented software prototype and development environment for the Spatial Grammar Machining Planning (SGMP) system is given.

1.6 Summary of research contributions

This thesis aims to make the following contributions:

- Development of a machining knowledge representation including geometry and semantics of machining operations that allows for direct integration with and application by machine tools (Sections 3 & 4).
- Development of a new bottom-up, process-based method for CNC machining planning in automated design-to-fabrication using spatial grammars and heuristic search (Sections 3 & 4).
- Application of spatial grammars and design synthesis methods to decompose designs into feasible machining plans (Section 4).
- Implementation of the method in a software prototype able to successfully plan CNC machining operations for a variety of parts (Sections 5.1, 5.2 & 5.3).
- Unique example of use of spatial grammars in a hardware-based spatial grammar implementation (Section 5).
- Powerful and extensible spatial grammar framework for CNC machining planning, able to
    - capitalize on newly added or changed capabilities instantly (Sections 5.4 & 5.5) and to
    - react online to changes of the hardware (Section 5.6).

A thorough discussion of these contributions with respect to the state-of-the-art is presented in Section 7.1.

2 Background

In the following the state-of-the-art in feature technology and its use in design-to-fabrication, namely feature recognition, is presented. Further, existing approaches to intelligent systems for planning and fabrication are reviewed. Rapid Prototyping and Rapid Manufacturing as competing technology for automated design-to-fabrication will be presented. Spatial grammars with their use in engineering are presented as powerful method to capture design knowledge and apply it to engineering problems. This section concludes with a brief summary of the state-of-the-art.

2.1 Feature technology

A driver for the use of features has been the desire to integrate design with further, downstream applications such as manufacturing and process planning (SHAH et al. 1995). A feature enables this by representing "… the engineering meaning or significance of the geometry of a part or assembly" and by offering the following characteristics (SHAH & MÄNTYLÄ 1995):

- A feature is a symbol that has a significant meaning within a defined context,
- it can be linked to a shape,
- features are the building blocks for a part and
- there is a common agreement about the feature's properties.

Consequently, the definition of a specific feature is arbitrary and context dependant (SHAH 1988), i.e. in different domains different features are used. To translate from one domain to the other a mapping is required. This mapping is rarely one-to-one, instead, the mapping is highly non-linear and complex. Due to this, several approaches were developed: Feature Mapping, Design-by-Features (DbF), and Feature Recognition.

Feature Mapping

Feature mapping attempts to connect different views of the same entity, each specified by features, by describing how features from the one view can be translated to the other view and vice versa. This mapping is required since feature descriptions can vary according to their level of abstraction, completeness of their description (e.g. information might be omitted), parameterization, what they describe (e.g. solids in design and voids in injection molding design) and application specific attributes (e.g. for process planning, electric wiring) (SHAH & MÄNTYLÄ 1995). Due to the widespread meaning of the term feature mapping, it is more of a general description of desired methods than a specific technique. For symbolic and rigid specification, a translation table of how to map features can be established. With increasing complexity and semantic meaning of features, prescribed automatic translation becomes infeasible. Instead, the mapping then requires knowledge about the relations of the features and their underlying geometry (SHAH et al. 1995). This is especially true, considering disjoint feature domains, e.g. design and manufacturing. Here, features from one domain may not have an appropriate counterpart in the other domain since they are not of interest there. An example might be an indexing surface for part locating. This feature is important in manufacturing but is insignificant in design (CUTKOSKY et al. 1988).

Design-by-Features (DbF)

In design-by-features, the user specifies the feature model using a feature library of valid features. The geometric model can then be automatically created using a geometric modeler. The features can either be constructive, i.e. the design begins with an empty space from which features can be added or subtracted, or destructive, i.e. the design begins with a stock from which features can only be subtracted (SHAH & MÄNTYLÄ 1995). In the case of destructive solid geometry design, the resulting part design is not only a shape but also a manufacturing plan (CUTKOSKY et al. 1988), if there is a mapping from the features to machining operations. However, it is impossible to fabricate the desired part using alternative ways (SALOMONS et al. 1993) since the working steps are fixed. This poses difficulties as manufacturing processes may change over the life-time of a design and optimal processes cannot be chosen due to the fixed manufacturing feature decomposition (VANDENBRANDE & REQUICHA 1993). Further, the designer must have a clear picture of the part he wants to design for which "… design by manufacturing features is not viewed appropriate for innovative design." (SHAH et al. 1994b).

Feature Recognition

In automatic feature recognition (FR) the user is replaced by an automated system that finds features within the geometric model (SHAH & MÄNTYLÄ 1995). More specifically, FR identifies the portions of the geometric model with a certain characteristic which is of interest for a specific application (SHAH et al. 2001). In the field of design-to-fabrication this refers to the identification of manufacturing features or machining volumes. SHAH et al. (2001) present an excellent review of current approaches to FR. Their findings are presented in the following.

While it poses little difficulty for a human to reason about complex geometry and to think about where to remove what volume to yield a desired part, the very same task is a hard problem for computer algorithms due to the following challenges (CORNEY et al. 2005). Several perspectives and different representations are required to capture the complete design intention within a CAD-System. This can be required from different disciplines such as mechanical or electrical engineer or reside within the same discipline such as general mechanical design and tolerancing. Multiple interpretations of the same part must be allowed to enable a broader range of solutions in later process steps. This can be multiple interpretations of a designed part in terms of manufacturing features for milling or as an alternative as die-casted part. At the same time, the information gap between different domains, e.g. the design (CAD) and the manufacturing (CAM), must be bridged which is even more difficult since design features do not necessarily directly map to manufacturing features. A design feature does not have a manufacturing meaning per-se and a manufacturing feature does not automatically have any function or design meaning. Also, features are context dependent. Features in conventional mechanical design differ strongly from e.g. sheet metal design features. In manufacturing, features for milling applications differ largely from features for turning applications, forging or die-casting. In each of these domains, features have to be identified and formalized which is labor intensive. Further, different abstraction levels of features and the inflexibility in their definition complicates this challenge (SHAH et al. 1994b).

Existing approaches to FR can be classified according to their underlying principle. Figure 2-1 shows the taxonomy divided by topological, heuristic, symbolic, volumetric, process-centric and hybrid approaches. The individual techniques are presented for the convenience of the reader based on SHAH & MÄNTYLÄ 1995 and SHAH et al. 2001.

Topological approaches use the information from the relationship of topological entities (edges, faces, points) of the geometric model. This information can be the existence of adjacent faces or the interconnections between a set of topological entities. Using a graph-based representation of all topological entities of the geometric model, the desired features can be identified by matching their graph of topological entities to a sub-graph of the geometric model.

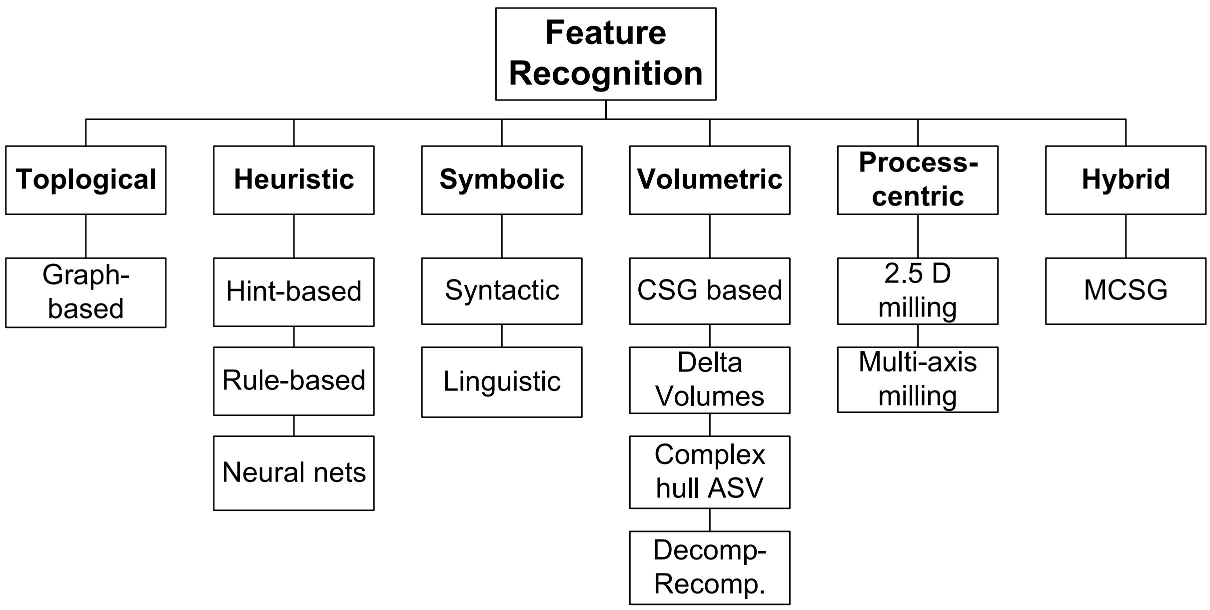


Figure 2-1: Taxonomy of Feature Recognition techniques (SHAH et al. 2001)

Heuristic FR approaches use guesses to identify potential features. Hint-based approaches generate such guesses from incomplete patterns in the boundary representation of the geometric model. The identified features are then extended in terms of size to their maximum volume. Through this, interacting features, i.e. features that lost or gained relations of their entities, can still be identified. Rule-based recognition uses formalized production rules to identify features. For each feature type, a separate rule has to be created and evaluated once for every feature which can be time consuming and thus prohibit the application to large models (SHAH & MÄNTYLÄ 1995). As advancement over classical rule based approaches, neural nets give a probabilistic estimation as to whether a feature exists.

Symbolic methods form the next group of techniques. Symbols can be used to describe patterns such as a series of line segments. Syntactic pattern recognition uses vision system methods to match these patterns. From the symbols, a grammar describing the sequence of symbols algebraically and manipulating them can be formulated, thus forming a language of valid patterns.

In contrast to topological recognition methods, volumetric recognition techniques rely on the existence of complete volumes. The search can be directly performed on the constructive solid geometry (CSG) tree by matching the feature definition to a sub-tree of the CSG tree. To accomplish this, the CSG tree must be rearranged to some canonical form as to enable the algorithm to successfully identify the desired features (SHAH & MÄNTYLÄ 1995). The Delta Volumes technique aims to identify the volumes on a workpiece stock that correspond to machining operations, thus identifying the delta volume on the workpiece for each operation.

Convex Hull Alternating Sum of Volumes (ASV) calculates the convex hull for a given part and then subtracts the part model itself from the convex hull. From the set of resulting disjoint volumes, the same algorithm is applied recursively, creating the convex hull for each disjoint volume and subtracting the volume from the hull until the convex hull matches the part's volume. Through this method either a decomposition of positive features (design features) or negative features (e.g. machining volumes) can be obtained.

Volume decomposition and re-composition splits up the removal volume (stock minus the desired part) into separate volumes. The volumes are then re-combined into different configurations using Boolean operations. This means, from a single decomposition, multiple possible re-combinations can be created. Through this, lost regions due to interacting features can be restored and multiple feature interpretations generated.

Process-centric techniques focus on the recognition of machining volumes using specific knowledge on the machining process. 2.5D milling sections the part to be machined horizontally into layers such that the outlines of the part can be obtained. Fitting toolpaths can then be created by offsetting the contours towards the non-material side and connecting the resulting paths. The technique has also been extended to allow for the toolpath generation for pockets with freeform bottom surfaces. Multi-axis milling requires a more complex recognition method. Based on a feature taxonomy of 5-axis milling features, recognition methods for each feature class based on geometric and topological characteristics have been designed and implemented (SHAH et al. 2001). However, through the use of a universally applicable plan for machining a part by cutting it layer-by-layer, efficiency and accurateness is lowered significantly (BOURNE et al. 2011).

Hybrid techniques take advantage of two or more basic FR techniques. One example is the Minimal Condition Sub-Graph (MCSG) method. To recognize features, an Attributed Adjacency Graph (AAG) of the parts faces is created. This graph represents all faces of the parts and their connections. From the AAG all connected faces that do not belong to the complex hull or the stock are extracted into the Manufacturing Face Adjacency Graph (MFAG). The faces in theses graphs must be created during manufacturing and thus represent valid machining features. Interactions between the features can be resolved by using hints from the MFAG. If the AAG of a feature represents the maximal sub-graph of the MCSG of a feature, i.e. it fulfills the conditions for being that specific feature, the feature is identified.

Despite the progress made, no single approach solves all the challenges in feature extraction (CORNEY et al. 2005). It rather seems that there is no single best technique for all types of features and applications (SHAH et al. 2001). In industrial practice, feature recognition has not been significantly implemented (RAHMANI & AREZOO 2007).

However, besides the challenge of feature recognition and feature mapping and the limitations of design-by-features, several limitations and difficulties already exist by using feature technology: The most fundamental limitation is that “the semantics of the data is not embodied in features” (SHAH et al. 2001). This means, a feature is only a symbol and therefore requires a mapping to actions or processes to give it meaning to something like a machine tool. The long emerging standard STEP-NC aims at offering a set of machining features which should enable machine tool vendors to program their products to automatically decompose each STEP-NC feature into feasible machining operations. However, having a machining feature representation of a part, already pre-defines the manufacturing process to a great extent. This can result in less efficient manufacturing due to the highly coupled planning problem (BOURNE et al. 2011) where seemingly each decision influences the ones already made and of course future ones as well. To overcome this, multiple feature interpretations can be used. This then leads to exponential growth in complexity of the planning problem, since multiple interpretations of features, through feature interaction, increase the number of possible combinations of operations (BOURNE et al. 2011). Also, feature technology requires maintaining a large number of specialized features due to the narrow scope and required exactness to allow for a successful mapping. Especially since features and feature recognition is not universal but depends on the intended application and the required level of detail: For NC machining planning, a lot of detail knowledge about tools and machine capabilities is required whereas in manufacturability analysis rough information is sufficient (SHAH & MÄNTYLÄ 1995). Also, the work to identify and formalize features for each domain is labor intensive (SHAH et al. 1994a). Even if a satisfyingly large set of features is available, each feature must be mapped to actions or processes to give it meaning, thus further increasing the amount of work to establish and maintain features and their mappings.

2.2 Intelligent systems for planning and manufacturing

In the last years, research moved from realizing single component systems for automated design-to fabrication towards more complete systems. These should then be able to produce complex parts while maintaining a high degree of automation or even autonomy during runtime. In the following, a number of selected systems are presented.

2.2.1 Smart Machining Systems

The approach of "Smart Machining Systems (SMS)" at the National Institute of Standards and Technology (NIST), as presented by DESHAYES et al. (2006) and NEWMAN et al. (2007), focuses on the optimization of machining processes during runtime and knowledge required to decide what to optimize in the detailed process plan and which tactics should be applied to the runtime optimization. Through this, a SMS shall be enabled to learn from experience, monitor and optimize its operations, assess the quality of its work and communicate its capabilities using a high level language. Through these capabilities, the detailed process optimization during production ramp-up should be avoided, the system should react to changes in demand and the overall manufacturing time should be shortened enabling a rapid manufacturing

The need for runtime optimization stems from using approximation models for the planning stage which still contain uncertainties. For the planning stage, various models for machine performance and cutting force prediction are used applying experience based models, machining experiment models and models based on standard material properties. To capture and represent the semantics and syntax of the required knowledge, ontologies are used.

As of 2006, the research was in its initial stage. The researchers see the most important issues in developing a unified methodology for integrating the different machining process models and in the necessary effort to develop the software enabling SMS. However the research focus within SMS is the optimization of processes during runtime of the system using sensor feedback and not the machining planning down to CNC code generation. Since SMS capitalize on the manufacturing feature representation STEP-NC, the limitations presented in Section 2.1 apply here as well.

2.2.2 Shandong University Intelligent CNC

Researchers at the Shandong University are working on an intelligent CNC controller. This CNC controller will be using STEP-NC to carry out process planning down to real-time control of the machine tool (ZHANG et al. 2006). To achieve this, the tool-path planning capabilities traditionally embedded within a CAM system shall be transferred to the CNC controller itself. The Intelligent CNC should use a STEP-NC compliant part program as input to carry out the detailed and process specific planning for the part manufacture itself (LIU et al. 2006). Because the system uses a STEP-NC part program, the fabrication process is already pre-defined to great extent. Only process-specific information such as feeds and speeds need to be planned on the CNC controller level. Similar to Smart Machining Systems (SMS), due to the use of STEP-NC, the same limitations, presented in Section 2.1, apply.

2.2.3 Research Cluster of Excellence "Integrative Production Technology for High-Wage Countries"

The Cluster of Excellence "Integrative Production Technology for High-Wage Countries" focuses on the research areas of "Individualized Production", "Virtual Production Systems", "Hybrid Manufacturing Technology" and "Self-optimizing Production Systems". In the following, selected research work on "Individualized Production" and "Self-optimizing Production Systems" will be further in detailed as it has been described by SCHUH et al. (2007).

Individualized Production

Research into Individualized Production focuses on the production of individualized, customized products. To achieve this research in Advanced Manufacturing Technologies for Individualized Products is underway. The work focuses on low-batch, possibly one-of-a-kind production of parts using modular dies and molds for die-casting and Selective Laser Melting. Especially the Rapid Prototyping process Selective Laser Melting promises to be suitable for the production of highly individualized functional metal parts. However, for Rapid Prototyping processes, several limitations exist that will be discusses in Section 2.3.

Self-optimizing Production Systems

Research into Self-optimizing Production Systems focuses on methods to enable production systems to analyze the current situation, assign and evaluate system objectives and to modify its behavior if necessary.

The subproject Cognitive Control Systems for Manufacturing Systems aims at creating "control technology that – for a given problem domain – is capable of aggregating distributed knowledge and of reasoning on this knowledge to detect interrelations between facts" (BRECHER et al. 2008). Through this, the system should be enabled to autonomously formulate subgoals, find solution paths, and initiate decisions under consideration of the current situation. The observation of the system over time should enable the system to cope with fuzzy, incomplete or contradicting information and to perform adequately and stable. The system architecture on the planning level relies on the SOAR architecture whereas the domain-specific knowledge is represented by a formal conceptualization similar but not identical to an ontology (BRECHER et al. 2008). The approach is applied to an assembly cell with two industrial robots. The approach is claimed to reduce the planning and programming effort of the industrial robots, especially if more resources with the same capabilities are added (KEMPF et al. 2009).

However, the focus of this Research Cluster of Excellence is the application of Rapid Prototyping Technology to individualized production and the self-optimizing of assembly operations and not the fabrication of parts using CNC machines.

2.2.4 CyberCut

The idea of CyberCut is to enable the rapid fabrication of parts similar, to the existing Metal Oxide Semiconductor Implementation Service (MOSIS) for the production of Very-large-scale integration (VLSI) circuits. The development of CyberCut stems from the Integrated Manufacturing and Design Environment (IMADE) and transforms it into a distributed agent environment (SMITH & WRIGHT 1996).

CyberCut integrates World Wide Web technologies with IMADE's CAD/CAP/CAM process chain creating a web-based design-to-fabrication system (SMITH & WRIGHT 1996). Offering a web-based CAD software to design in destructive machining features, the process planning system is able to create the macro- and micro-plans for the fabrication of the parts iteratively, giving feedback to the user on the manufacturability of the part right in the design phase (AHN et al. 2001). The created process plans can then be executed on a 3-axis milling machine which is integrated in CyberCut. The process planning also allows setup and fixture planning for conventional vices, toe clamps or Reference Free Part Encapsulation (RFPE) (AHN et al. 2001).

The initial CyberCut system was limited to 2.5D prismatic milling parts which had to be designed using Destructive Solid Geometry, i.e. design-by-features, which allows for easier mapping of machining features to machining operations (AHN et al. 2001). However it has been extended to allow machining of freeform surfaces using 3-axis milling as presented by WRIGHT et al. (2004). SUNDARARAJAN & WRIGHT (2007) extended the system with Feature Recognition techniques to allow, in parallel to the web-based, specialized CAD software, the use of a Boundary Representation input to the CyberCut process planning pipeline.

However, a number of strong limitations do exist (SUNDARARAJAN & WRIGHT 2007): The fabrication capabilities are limited to a single generic 3-axis milling machine. The stock has to be cuboid to fabricate the part and it must be the bounding box of the part. The feature recognition algorithm is rather simplistic. Because of this, features can be only extracted from six orthogonal directions and features with undercuts such as dove-tails are not recognizable. Each recognized feature is represented by its planar contour, access direction, depth of the feature and whether the feature goes through the whole stock. Because of this specialized feature representation, the approach can only be applied to 3-axis milling.

2.3 Rapid Prototyping and Rapid Manufacturing

Rapid Prototyping (RP) and Rapid Manufacturing (RM) technologies can avoid the issues of CAD/CAM and feature recognition by using a fundamentally different approach to the fabrication of parts that reduces the complexity of fabrication planning. A review of RP technologies is now discussed and related to automating the CAD/CAM interface.

The relatively young disciplines of RP and RM have many different definitions. First, it is necessary to differentiate between rapid prototyping technology and its applications (GEBHARDT 2003). On the technology side, RP can be regarded as a means of Layered Manufacturing (LM) (LEVY et al. 2003). On the application level, rapid prototyping techniques can be defined as a group of technologies capable of creating a 3D solid object with little or no human intervention once the manufacturing process has begun (ONUH & YUSUF 1999). To some extent, CNC cutting, which traditionally is not regarded as a RP technology, can be included in this definition (BURNS 1993). Nevertheless, direct component production (LEVY et al. 2003), where a part is directly produced from the corresponding geometry file, is common to these definitions, while CNC machining still violates the condition of little human intervention in practice.

Rapid Manufacturing can be defined as super-set to rapid prototyping and rapid tooling (RT) (PHAM & DIMOV 2001). RM can also be viewed as an application of RP to produce end-use parts from a rapid prototyping technology (GEBHARDT 2003).

Comparing LM to CNC machining, it is possible to conclude (GEBHARDT 2003) that with CNC machining almost every material can be processed, whereas LM is bound to certain materials depending on the rapid prototyping process, which sometimes only mimics the original material characteristics. This is especially true for producing metal parts using RP technologies. Since the metal must be deposited in a controlled way, depending on the RP process, different methods are used. Most common are depositing melted metal (e.g. heated by a laser), metal powders glued by a binder, metal sheets welded together or metal powder within a paste based material. Common to the processes is the required post-processing such as furnacing of the part and infiltrating the porous part with another low melting point metal. Therefore, metal parts created using RP technology cannot reach the same strength as parts that were machined from the same material.

Further, in CNC machining almost every scale of parts can be manufactured with rich feature detail, high dimensional accuracy and good surface finish. In LM, the resolution of features is limited as well as the size is constrained by the machine envelope. On the other hand, LM has a couple of advantages over CNC machining. In LM the production time of a part is independent of the part-complexity whereas in CNC machining complexity can require additional setups, tool changes or can even prevent a part from being machined at all. Finally, CNC machining requires highly trained professional personnel. Even during operation, the attendance of a skilled craftsman is required to ensure safe and reliable operation of the machine tool. In contrast to this, some RP technologies can be used by non-experts, e.g. 3D printing or fused-deposition modeling.

In summary, automating the design-to-fabrication process for CNC machining will enable it to compete with RP technologies, especially for the direct production of end-use metal parts, where RP technologies have not succeeded to continue the success story of RP

(KOCHAN et al. 1999) and still have several issues (BERNARD & KARUNAKARAN 2008). The real breakthrough for RP will still depend on improving cost and productivity as well as improving material properties, accuracy and reliability (LEVY et al. 2003).

2.4 Spatial grammars and their applications in engineering

In the following, spatial grammars, i.e. grammars that work on spatial entities, are introduced, followed by a review of their use in engineering.

Production systems are a method used in cognitive systems and artificial intelligence (cp. Section 1.3) to solve specific problems, especially ill-structured problems. Such an ill-structured problem lacks quantitive descriptions, clearly defined goals and computationally exact methods to solve them (SIMON & NEWELL 1958) and comprises a complex dynamic environment with interactions and unpredictable and uncontrollable system behavior.

Consisting of a storage (production memory) for rules (productions), a set of symbols (working memory), and symbols (elements), production systems (KLAHR et al. 1987) form a generative mechanism, generally applicable to a wide variety of symbolic problems.

Formal grammars and algebras are such systems. While formal grammars work mostly symbolically, spatial grammars work in the domain of geometry and space.

According to HOISL & SHEA (2011), spatial grammars are a superset to different grammars that are able to represent spatial problems directly. The most fundamental type is shape grammars, a concept initially developed by STINY & GIPS (1972). After the refinement of the method in STINY (1980), a simpler form of shape grammars, the set grammar was developed. Instead of manipulating a shape and its components directly, set grammars work on sets of shapes (STINY 1982).

The advantages of spatial grammars lie within the same generation and analysis capabilities as production systems while being able to represent knowledge about both form and function and representing geometric shapes in addition to symbols (AGARWAL & CAGAN 2001).

A shape grammar, as well as a spatial grammar, consists, according to STINY (1980), of

- S – a finite set of shapes,
- L – a finite set of symbols,
- R – a finite set of shape rules,
- I – the initial labeled shape.

For the shape grammars, the algebras U_{ij} and V_{ij} define the subshape relation, operators and Euclidian transformations (STINY 1991) of shapes and labeled shapes respectively whereas i

refers to the dimension of the shape and j to the dimension in which the shapes are manipulated. These algebras do not only include the elements of the shape grammar and their transformation but also operators as for addition or subtraction (STINY 1991) or even more complex operations (SHIRUR et al. 1998).

The two main issues in implementing shape grammars, which currently prevent a full featured implementation of a shape grammar system, are (a) the generic representation of all shapes and (b) the recognition of emergent shapes. These two problems go hand in hand since the representation of shapes lays the foundation for the recognition of new and prior non-explicitly represented shapes. While humans have no trouble in recognizing alternative interpretations of shape by "simply seeing it" (STINY 2006), computers lack this ability. The issues are more difficult if shapes from different algebras interact, such as a planar shape is cut into two new shapes by a line. KRISHNAMURTI & STOUFFS developed a method and algorithms for calculating the shape boundary and thus representing shapes enabling emergence (KRISHNAMURTI & STOUFFS 2004, STOUFFS & KRISHNAMURTI 2006).

In contrast to shape grammars, set grammars are more convenient to be used in an algorithm (STINY 1982). HEISSERMAN & WOODBURY (1993) use in their solid model grammar a topology graph to make all entities of the solids accessible to a computer implementation. If only set grammars are considered, existing solid shape implementations such as CAD kernels can be used. Such an implementation of a spatial grammar system is presented by PIAZZALUNGA & FITZHORN (1998). Important to note is the fact that such a CAD kernel also supports shapes of different dimensionality but is not able to represent interactions of the different algebras nor does it enable the reinterpretation of shapes in order to support the subshape detection as shape grammars do. However, shapes from different algebras can exist in the same model.

In contrast to knowledge-based and case-based design systems or expert systems in general, spatial grammars comprise deep knowledge especially on geometry and geometric relations. Therefore, shape grammar based systems have proven to be able to represent and generate designs for numerous engineering applications:

AGARWAL et al. (2000) developed a shape grammatical system able to represent and generate Microelectromechanical Systems (MEMS). It uses a two-dimensional parametric shape grammar with weights on geometric elements and also considers plane surfaces and their boundaries. The grammar is implemented using LISP. Therefore, the system is limited to creating MEMS designs in a two-dimensional space. PUGLIESE & CAGAN (2002) developed a shape grammar to capture the stylistic formalism of a specific motorcycle brand and to create new motorcycle designs within the same stylistic formalism. However, the system only considered visual aspects without functional aspects and was not implemented.

MCCORMACK & CAGAN (2002) presented a shape grammar implementation for two-dimensional straight-line design problems and applied it to the automatic design generation of automotive hood panels. The work on the shape grammar implementation was generalized to also handle curves (McCormack & Cagan 2006). However, the system only considers two-dimensional shapes.

Grammatical approaches have also been successfully applied not only to design problems but also to the design-to-fabrication interface. Preiss & Kaplansky (1985) proposed using a production system, i.e. symbolic grammar, with rules that encode the machining operations and using heuristic search methods to generate machine tool instructions from 2D drawings for 2.5D milling. To create machining instructions the user is prompted to input the height of the different areas of the part. Therefore, the approach required the existence of an engineering drawing, i.e. it is not able to directly work on three-dimensional geometries and further interaction with an expert is required. Sass (2008)developed a system using a shape grammar approach to section a larger design model into smaller pieces that then fit into the envelope of the RP machine that is used to fabricate these parts. However, the primary means of fabrication is Rapid Prototyping and generating the instructions for the fabrication machines is not part of their research. A previous work of other researchers (Wang & Duarte 2002) already use a shape grammar framework to create designs and Rapid Prototyping technology to produce these parts. However, the system is limited to a small set of rules and therefore the design capabilities are limited as well. Also, the generation of the detailed fabrication instructions for the Rapid Prototyping machines is not part of their work.

Brown et al. (1995) use a feature grammar to represent valid, 2D designs for turned shafts. With this system, a user is able to design shafts in a design-by-features manner while receiving feedback on manufacturability from the computer. The work was later extended to include a generative search method (Brown & Cagan 1997) that enables automated process planning for turning of the designed shafts, based on the prior representation. However, this work is limited to two-dimensional rectilinear shaft designs and the process plan created cannot be directly executed on machine tools.

Ostrosi & Ferney (2005) use a feature graph grammar on topological and geometric graph representation of a part for feature modeling and feature recognition, especially the recognition of canonical and interacting features. In their work, features are recognized from faces based on the structure they are embedded in. Due to the nature of their approach, it suffers from similar limitations as feature recognitions approaches discussed in Section 2.1. Also, their work does not include the creation of production or machining plans or generation of detailed machining instructions.

The generative power of spatial grammars has been successfully combined with heuristic search methods to produce optimally directed functional designs by Cagan & Mitchell (1993). The method has been successfully applied to the design of three-dimensional truss structures (Shea & Cagan 1999). However, the approach is limited to generating designs only, without considering the fabrication of the truss structures. Starling & Shea (2005) developed a parallel grammar to represent valid configurations of gear systems and multi-objective optimization to generate gear systems according to a given specification. This approach focuses on the functional and spatial design of gear systems and not the fabrication.

2.5 Summary

Although much progress has been made in the last decades in the area of integrated CAD/CAPP/CAM, no single system provides the possibility to autonomously derive all necessary fabrication information down to the Numerical Control (NC) programs from a given 3D model. This is generally due to using a feature-based approach. However, the fundamental flaw of features, as discussed in Section 2.1, is that the semantics is not part of the feature but lies in the eye of the observer. Certainly, great successes have been realized using feature technology and feature recognition, however todays industrially used CAD/CAM systems are still aids to a sophisticated process planning expert and NC part programmer.

The efficiency of fabrication has always been of importance in production research. Consequently, researchers continuously strive towards more efficient production especially with the change of markets towards customized products and small batch production. The intelligent systems presented in Section 2.2 aim mostly at having the machines optimize their behavior themselves during runtime or focus more on assembly processes rather than machining processes. If the system should also integrate machining capabilities, the researchers rely still on the shallow knowledge representation of features. Building an intelligent or cognitive system using such a shallow knowledge representation might not lead to success.

Rapid Prototyping (presented in Section 2.3) seemed to be a loophole to overcome the deficits of CAD/CAM systems, providing a uniform and rather simplistic preparation process for the fabrication of arbitrarily shaped parts. However, RP technologies are today still expensive and cannot perfectly compete with traditional machining when it comes to available materials, precision and strength of the parts fabricated.

New methods for an autonomous and flexible design-to-fabrication process are required that equip machine tools with knowledge about their capabilities, allow them to plan their own actions and thus make them more intelligent, beyond the traditional automation technology which is not intelligent at all (BRECHER et al. 2008).

Spatial grammars have proven to be able to capture domain specific knowledge as discussed in Section 2.4. Together with heuristic search methods this captured knowledge has been successfully applied to a variety of engineering problems.

The deep knowledge representation of spatial grammars for capturing the fundamental knowledge of specific machining processes and the power of heuristic search methods to apply this knowledge to generate feasible machining plans can be an enabler for automated design-to-fabrication of customized parts. Such an approach is presented in the remainder of this thesis.

3 Generative design and fabrication using shape grammars

Generative design and fabrication refers to the ability to autonomously generate designs while simultaneously generating all information to directly fabricate them. This technique is driven by the increasing need to rapidly and flexibly fabricate customized parts and individually designed products. For the automation of the design-to-fabrication process chain, intensive and dynamically updated knowledge from the domains of design and fabrication must be provided. To allow for a flexible, autonomous fabrication, the knowledge modeled must dynamically reflect the state of the fabrication system and its capabilities. This section presents an approach to unify knowledge for generative design and generative fabrication using shape grammars. With shape grammars, the geometry of designs and their mapping to removal volumes corresponding to fabrication processes on CNC machine tools are represented. The process instructions for fabrication are included by augmenting the removal volume shapes with labels. A new shape grammar approach to represent designs and fabrication processes is presented and validated on an example functional part as a proof-of-concept. The approach enables pushing knowledge downstream, from design and process planning directly to the fabrication system itself providing a stepping stone towards autonomous fabrication planning for customized parts.

3.1 Method for generative design and fabrication using shape grammars

In the following section, a new method is presented for generating designs and automatically generating fabrication information for CNC 3-axis-milling using shape grammars. Further, it is illustrated that the gap between the domains of design and fabrication can be bridged with shape grammars. To accomplish this, shape grammars are used to represent design and fabrication information. The approach presented covers the steps of design generation, process planning and fabrication.

A customized toy maze is chosen as a proof-of-concept demonstration part for a functional design. It allows the direct generation of numerous variants with a small underlying shape and rule set. Given a designable region and constraints on pocket size and depth, unique mazes are generated automatically. Four holes are added on the edges to allow for a cover bolted onto the maze from the top. Further, only macro-geometric aspects are considered, allowing focus on how to create and fabricate shapes rather than considering detailed aspects including machining process stability and tolerances.

3.1.1 Design generation using shape grammars

The Maze Grammar (MG) describes a maze design by the volume to be removed and allows therefore a direct mapping between the geometry and the necessary processes to manufacture the part. The maze layout is constrained to form a perfect maze, i.e. it has no loops, unreachable or open areas. The elements of the grammar are subject to a V_{03} x V_{33} algebra, however to yield working mazes the transformations are further constrained. For ease of visualization, the figures of the Maze Grammar are shown only in top view.

The labels and shapes of the maze grammar are shown in Figure 3-1.

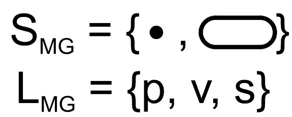


Figure 3-1: Shapes and labels of the Maze Grammar (MG).

The set of shapes consist of a point and a pocket representing the volume to be removed to connect two cells of the maze. The set of labels consist of the label *p*, denoting the current position of the maze growth, the label *v* to denote a point being unoccupied by a pocket and the label *s* indicating a point to be occupied by a pocket.

To create feasible and functional designs, a regular and rectangular point grid with a uniform grid size of ℓ is created within a defined boundary (Figure 3-2). This is the initial shape for the Maze Grammar.

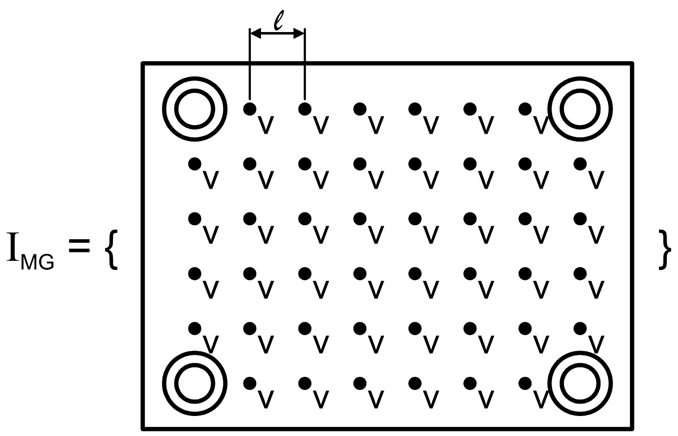


Figure 3-2: Initial shape of the Maze Grammar with point grid and holes.

Figure 3-3 shows the rules for the creation of the maze itself. The first rule picks a random point within the grid and initially sets the label *v* of the point to the label *p,s* denoting the point to be occupied (label *s*)and that the point is now the current maze growth position (label *p*). The second rule connects two adjacent points labeled *p,s* and *v* by a pocket of the uniform grid length *ℓ*, width *w* and depth *d*. Further the label *p* is moved to the point formerly labeled as *v* yielding the label *p,s*. The third rule transfers the label *p* between two arbitrary adjacent or non-adjacent points labeled *s*. This rule changes the position of the maze growth. The fourth rule is the terminal, finally eliminating the label *p* to stop maze growth.

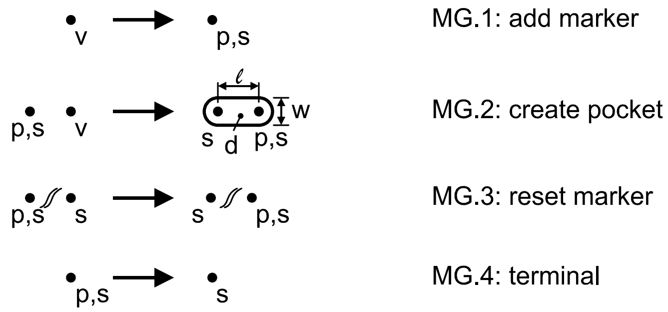


Figure 3-3: Maze grammar (MG) rule set.

As possible transformations to these rules, only translations and rotations by multiples of 90° within the same plane are allowed.

3.1.2 Shape grammar for CNC machining

After applying shape grammar rules to generate designs, shape grammars can also be used to reason about geometry and be applied to fabrication process planning. To bridge the gap between design and fabrication, intensive knowledge of the fabrication process is required. This thesis focuses on CNC controlled machine tools for cutting as a means of fabrication. Nevertheless the approach is intended to be general and suitable for alternative fabrication processes.

For machining, in this case CNC controlled 3-axis milling, knowledge on how to instruct the machine tool, the semantics of commands and the tool-workpiece interaction and constraints, must be provided. To meet these requirements, a two-layer grammar framework shown in Figure 3-4 is proposed.

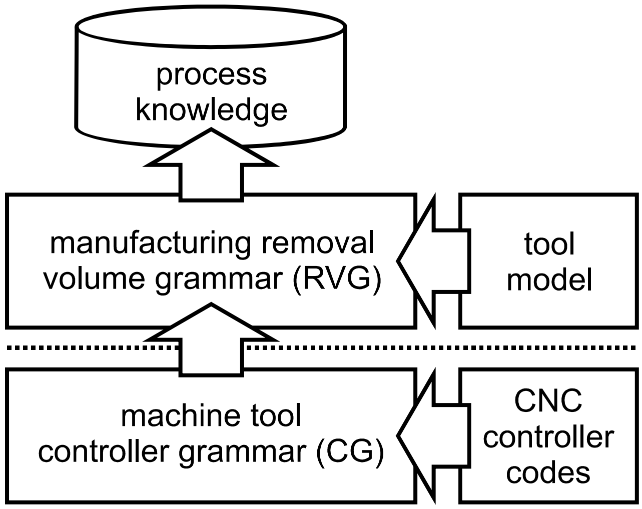


Figure 3-4: Two-layer grammar framework for removal volume and CNC code generation.

On the lower level, the CNC controller codes, which encode the tool movements and machine parameters, consisting of a specific command and its parameters, are mapped to toolpaths. The mapping is done by the Machine Tool Controller Grammar (CG). The elemental toolpaths or tool trajectories are then used in the Manufacturing Removal Volume Grammar (RVG) to generate the removal volume, considering the tool shape and constraints on cutting conditions provided by the tool model. Together, the grammars model a set of valid CNC operations and syntactically valid CNC controller code. The gathered process knowledge is the output of this framework.

Figure 3-5 shows the inputs and output of the Machine Tool Controller Grammar (CG). The input is the CNC machine instructions and syntax, i.e. the controller specific commands. As output, the CG generates a sequence of CNC machine operations with toolpaths and the corresponding CNC code.

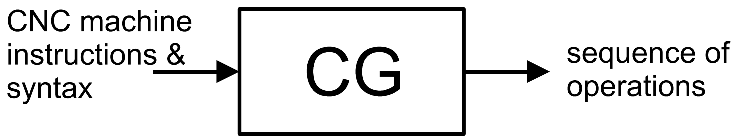


Figure 3-5: Input and Output of the Machine Tool Controller Grammar (CG).

The Machine Tool Controller Grammar (CG) has the shape set S_{CG}, consisting of a point, a line and an arc. The labels are *m* to denote the current position of the virtual tool tip, the label *n* to indicate a point being member of the CG, the coordinate label *(x,y,z)*, *r* for the radius of an arc and the labels *G1*, *G2* and *G3* for the CNC G-commands straight line interpolation, clockwise circular interpolation and counter-clockwise circular interpolation. The shape set and labels are shown in Figure 3-6.

$$S_{CG} = \{\, \cdot\,, —, \frown \,\}$$
$$L_{CG} = \{m, n, (x, y, z), r, G1, G2, G3\}$$

Figure 3-6: Shapes and labels of the Machine Tool Controller Grammar (CG).

Figure 3-7 shows the rules of the CG. The first rule creates a point at an arbitrary spatial position, labeled *n* for being member of the CG and *m* as current position of the virtual tool tip.

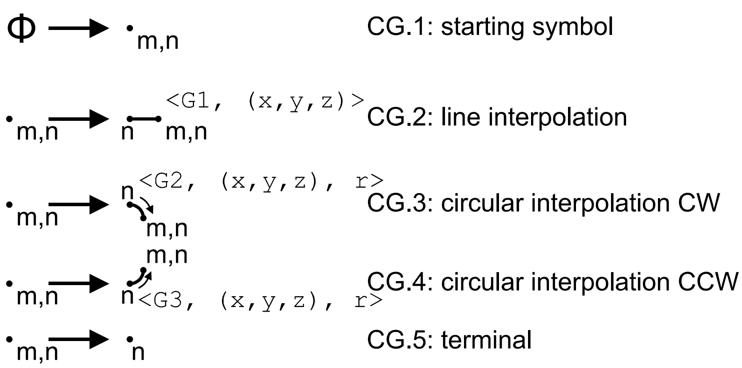


Figure 3-7: Rule set of the machine tool controller grammar (CG).

The second rule creates a straight line interpolation from the current position of the virtual tool tip and moves the label *m* to the newly created point at the end of the line. At the same time, the CNC command *G1* and the coordinates of the newly created point are attached to this toolpath. The third rule creates a clockwise arc and moves the label *m* analogous to rule CG.2. The label *G2* along with the coordinate 3-tupel of the arc's endpoint and the arc's radius *r* are attached to the toolpath. The fourth rule works similar to rule CG.3, only it creates a counter-clockwise arc and attaches the *G3* command to the toolpath. The fifth rule terminates the toolpath generation by removing the label *m*.

As transformations of the algebra V_{03} *x* V_{13}, all translations and rotations as well as scaling can be applied. Mirroring is not allowed since it would change a clockwise arc to a counter-clockwise arc and vice-versa.

Figure 3-8 shows the inputs and output of the Manufacturing Removal Volume Grammar (RVG). The grammar uses the tool shape and constraints, e.g. cutting directions, and the sequence of operations with their toolpaths from the CG and generates the removal volume along with the CNC code used for the creation of the removal volume.

Figure 3-8: Inputs and output of the Manufacturing Removal Volume Grammar (RVG).

Figure 3-9 shows the labels and shapes of the Removal Volume Grammar (RVG).

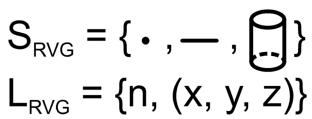


Figure 3-9: Shapes and Labels of the Removal Volume Grammar (RVG).

The set of shapes consist of a point, a line and a cylinder representing the tool shape, in this case an end-mill. The set of labels consists of the label *n* identifying points from the Controller Grammar (CG) and the coordinate label *(x,y,z)*.

Figure 3-10 shows the rules of the Removal Volume Grammar.

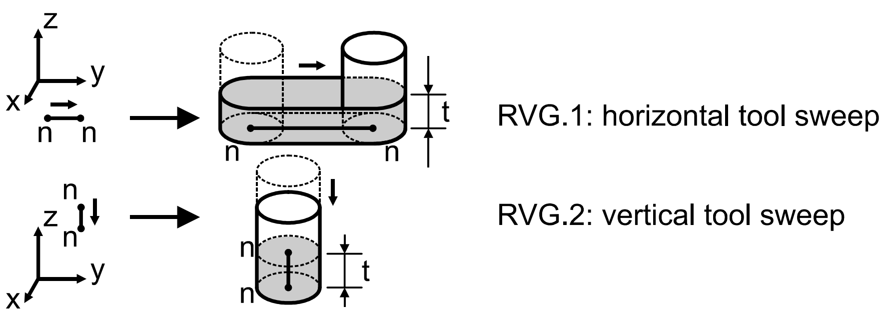


Figure 3-10: Rule set of the Removal Volume Grammar (RVG).

The first rule transforms a horizontal straight line toolpath into a removal volume by sweeping the tool shape along the trajectory of the toolpath. The removal volume is constrained by the maximum cutting depth *t* of the tool. The second rule creates a cylindrical removal volume (with the height *t*) from a vertical straight line toolpath analogous to the first rule. The CNC commands used to create the removal volume (not shown in Figure 3-10) are inherited from the Controller Grammar.

It is important to mention that the applicable transformations to the single shape elements reflect the kinematic properties of the machine tool. For a 3-axis-milling machine, the algebra V_{03} x V_{13} x V_{33} is restricted to translations in x- and y-direction and rotations around the z-axis. Further, any scaling, skew and mirroring is not allowed.

3.1.3 Mapping from design to fabrication

To overcome the differences between shape-based design and process-based fabrication, several mapping solutions are possible as shown in Figure 3-11.

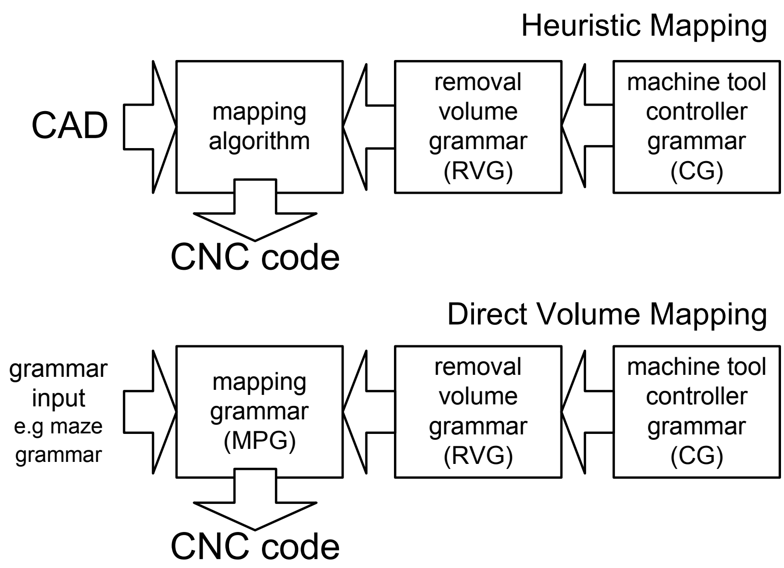


Figure 3-11: Mapping types for the sequential overall process.

Using feature recognition in process planning, the preferred method is to use a heuristic approach to map removal volumes of a process to the target removal volume represented in a CAD model (Figure 3-11 top). Here, a direct volume mapping is used by mapping removal volumes of machining sub-processes or operations to portions of the shapes created by the design grammar, in this case the maze grammar (MG), until the desired shape is fabricated.

The Mapping Grammar (MPG) links the design grammar to the fabrication process. It integrates the RVG and CG as described before. The inputs and output of the MPG are shown in Figure 3-12. The process constraints, e.g. an uninterrupted tool trajectory, and manufacturing information are inherited from the underlying RVG and CG. Through this, the complete CNC code instructions are generated by applying the MPG rules.

Figure 3-12: Inputs and output of the Mapping Grammar (MPG).

Figure 3-13 shows the shapes and labels of the MPG.

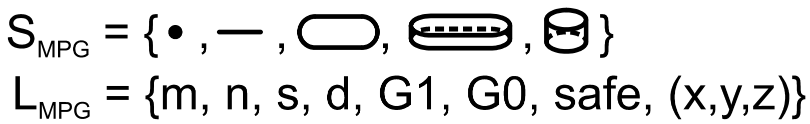


Figure 3-13: Labels and shapes of the Mapping Grammar (MPG).

The set of shapes contains a point, line, the pocket from the MG and the horizontal and vertical removal volume generated by the RVG. The labels are *m* for the current position of the virtual tool tip, *n* to denote a point of the CG, *s* to denote a point of the MG, *G1* and *G0* as machining commands for straight line interpolation and rapid feed positioning, *safe* to indicate to move the virtual tool tip to a position from which it can be moved in rapid mode without collisions and the coordinate 3-tupel *(x,y,z)*. A *safe* position in 3-axis-milling usually refers to a certain z-plane above the workpiece.

The Mapping Grammar (MPG) links the designed shapes to specific machining processes. Further the grammar considers physical constraints on accessibility and possible collisions between tool and workpiece.

Figure 3-14 shows the mapping rules where the shapes and labels in the left-hand side are the result of the maze grammar (MG), or design representation.

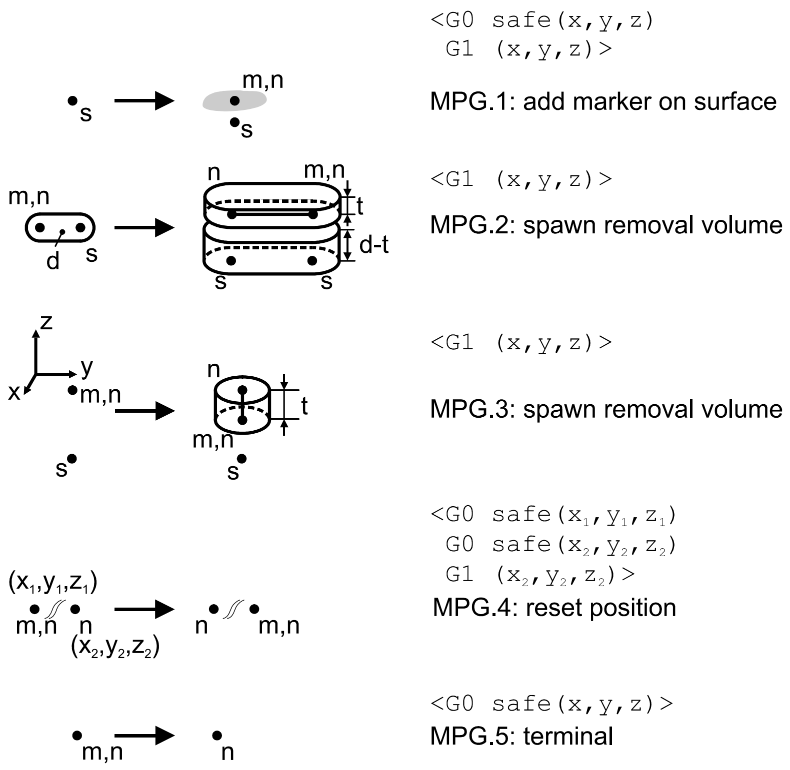


Figure 3-14: Rule set of the Mapping Grammar (MPG).

The first rule sets the virtual tool tip labeled *m,n* on a surface of the workpiece near to a point labeled *s* of the MG. Simultaneously, CNC commands to approach the new position of the virtual tool tip are generated. First, a safe position near to the target point is approached at rapid feed, avoiding collisions, and then the final position is reached at work feed. The second rule spawns a horizontal removal volume of the thickness *t* at the position of a design-pocket of the depth *d*. After applying the rule, the virtual tool tip is moved to its new position and the depth of the design-pocket reduced by the amount of *t*. The gap between the removal volume of the tool sweep and the residual volume is introduced for graphic display only. The CNC command *G1* along with the destination coordinates is generated at the same time. The third rule creates a removal volume by a vertical tool sweep of the height *t* at the current virtual tool tip position vertically aligned with the point labeled *s* of the MG. After applying the rule, the virtual tool tip position is moved and the CNC command generated. The fourth rule allows the virtual tool tip to return to the position it has been previously at. To do so, the tool is retracted to a safe position, near the first point at rapid feed, i.e. *G0* command, then moved to a *safe* position near the second point at rapid feed and then lowered to the final destination at work feed. The fifth rule removes the virtual tool tip position and retracts the tool to a safe position near its last position.

To explain further the process of mapping the design view into the fabrication process, Figure 3-15 shows an example of how a single pocket of the maze is mapped to a process and, at the same time, the CNC machining instructions are generated. The label *n* is suppressed for clarity.

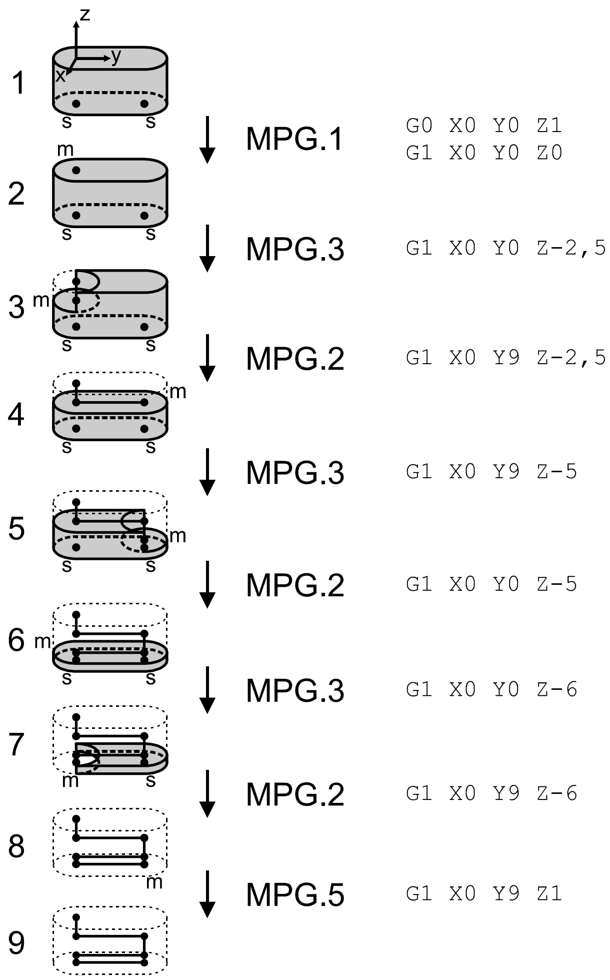


Figure 3-15: Example of the mapping between designed shape (gray) and manufacturing removal volume.

The first picture of the sequence shows the volume of the designed shape (gray) and the coordinate system used in the example. In the first step, the virtual tool tip is set onto the top surface of the pocket by applying rule MPG.1. From this, rule MPG.3 is applied to remove a cylindrical volume from the pocket's volume. Then, rule MPG.2 is applied to remove a horizontal slice of the pocket. In the following steps, the rules MPG.3 and MPG.2 are applied alternating until all of the pocket's volume is removed in step eight. Finally, the virtual tool tip is removed by rule MPG.5 to terminate the machining sequence. The generated CNC code can be executed on any standard 3-axis-milling machine under consideration of appropriate cutting conditions and coordinate system transformations.

3.2 Validation

After presenting the approach and method to generative design and fabrication, the Maze Grammar will now be validated. To allow for a more visual calculation of the grammar, the symbols in Figure 3-16 will be used instead of the labels.

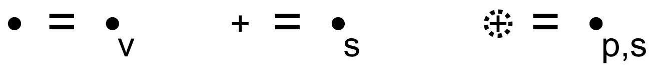


Figure 3-16: Symbols for visual calculation of the Maze Grammar.

The maze generation starts with the initial shape shown in Figure 3-17.

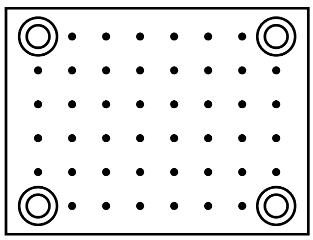


Figure 3-17: Initial shape of the maze.

First, the starting symbol is added (MG.1) to a random location, yielding Figure 3-18:

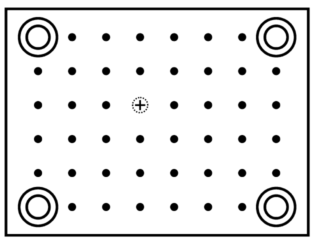


Figure 3-18: Shape of the maze after applying the starting symbol.

Subsequently the pocket creation rule (MG.2) is applied three-times in Figure 3-19 a-c.

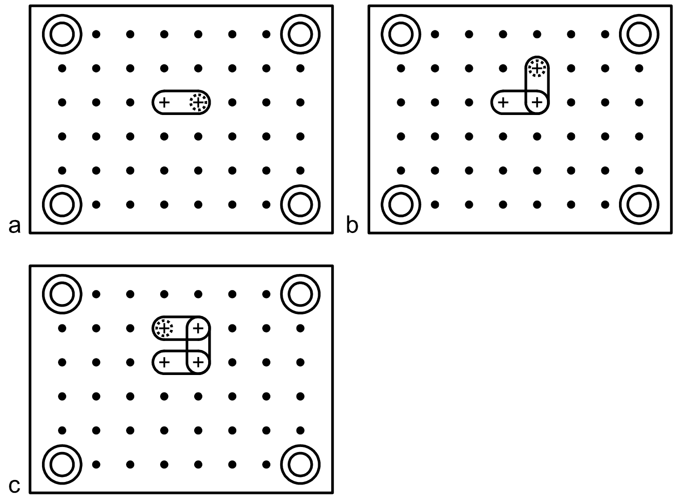


Figure 3-19: Creation of three pockets within the part.

After the first three pockets, the marker's position is reset (MG.3) and the creation of pockets (MG.2) is continued (Figure 3-20 a-d).

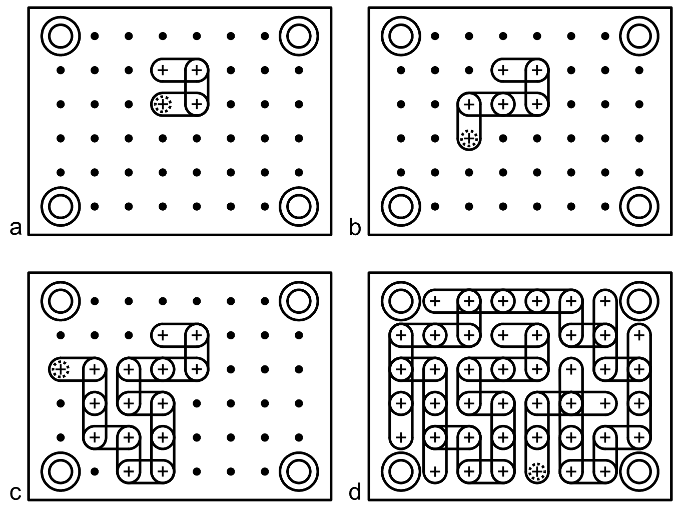


Figure 3-20: Further creation of pockets.

Finally the generation of the part is ended by applying the terminal rule (MG.4), thus removing the marker from the part as shown in Figure 3-21.

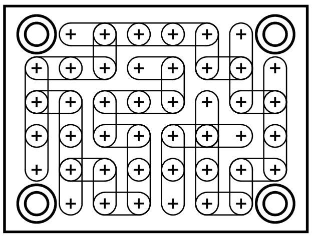


Figure 3-21: The maze after applying the terminal rule.

The final output of the shape grammar is shown in Figure 3-22.

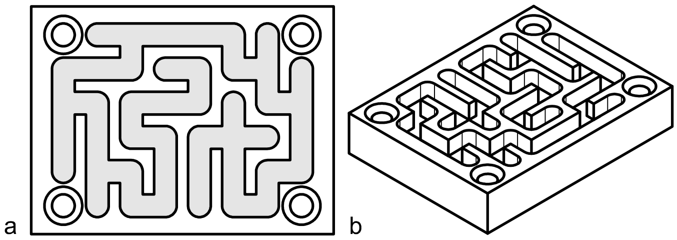


Figure 3-22: Output of the Maze Grammar (MG).

With the given design in Figure 3-22, the CNC code can be created by applying the Mapping Grammar to the design shown in Figure 3-21.

3.3 Discussion

In this example, the mapping between the design and fabrication domains is simple, yet demonstrates the main issues and potential for shape grammars.

With the process-based approach to representing knowledge in the grammar, machine tool capabilities are described and through application of grammar rules, process plans are produced for a desired design. Implicitly, the set of manufacturable shapes are represented by the grammar as well.

Different product variants can be generated by changing the initial shape of the grammar to different dimensions or shapes. While the maze generation using the Maze Grammar (MG) can be interrupted by applying the terminal rule at any time, resulting in a non-finished and possibly inoperable design, the mapping of the design to removal volumes must be continued until no rule from the Mapping Grammar (MPG), except for the terminal, can be applied. Otherwise, collisions between tool and workpiece can occur during the final retraction of the tool at the end of the CNC program. Further, prematurely stopping the MPG can result in only a portion of the design being fabricated.

The mapping grammar (MPG) only works with designs that are generated from the maze grammar (MG). Through this combination of grammars, an integrated representation for both design and fabrication is used. The advantage of shape grammars in the domain of CAD/CAM integration are that they provide a computable representation of the design geometry along with labels indicating fabrication information and are not restricted to a fixed feature decomposition or feature library. Further, the design-to-fabrication process can be transformed into a parallel process by alternating the application of rules from the Maze Grammar and the Mapping Grammar resulting in a concurrent process enabling the integration of fabrication knowledge in the design generation itself. However, this requires reducing generality, i.e. being forced to generate designs using grammars, to eliminate the need to recognize machining features from general CAD models. The rest of the thesis considers extending the initial approach to cover both, more complex shapes and handling general CAD models as input.

3.4 Conclusion

This section presented a shape grammar approach for generating both valid designs of a certain product class and their corresponding fabrication information to enable rapid fabrication of customized parts. The method provides an integrated representation for both generative design and fabrication. It was validated using an example of designing and fabricating customized mazes as a proof-of-concept. An important aspect of the approach is the representation of fabrication process knowledge in the Machine Tool Controller Grammar, Removal Volume Grammar and Mapping Grammar covering the capabilities of machine tools and tool shapes. However, the approach relies still on a design generated using a shape grammar to allow for a successful mapping to fabrication processes. This limitation is addressed in the next section.

4 Generative machining planning using spatial grammars and heuristic search

The previously presented method is limited by requiring the designer to express the design in a shape grammar formalism. To give the designer more freedom in the design expression, a method, based on spatial grammars and heuristic search that allows for creating machining plans from 3D CAD geometries is presented. To create such plans, domain specific knowledge is required to map the desired geometry of a part to a manufacturing process, thus decomposing design information into a set of feasible machining operations. Approaches to automating this planning process still rely heavily on human capabilities, such as planning and reasoning about geometry in relation to machining capabilities. To avoid the use of static feature sets and their pre-defined mappings to machining operations, the method encodes knowledge of fundamental machine capabilities. A method for generating a vocabulary of removal volumes, based on the available tool set and machine tool motions, is defined in combination with a basic rule set for material removal, covering tool motion, removal volume calculation and CNC code generation. The use of spatial grammars as a formalism in the Spatial Grammar Machining Planning (SGMP) method enables systematic formulation of constraints on spatial relations between the volume to be removed and the removal volume shape for a machining operation. A prototype implementation of the core method is presented.

4.1 Process for general design-to-fabrication

The process to translate a general 3D part model into a machining process plan in terms of CNC machining instructions is shown in Figure 4-1. It begins with the 3D geometric models of the part design and the stock (workpiece). The Total Removal Volume (TRV), i.e. the volume, which needs to be removed from the stock, or workpiece, to yield the finished part, is generated by applying a Boolean subtraction operation. The knowledge of the available machining processes can be generated from the available tools and machine tool capabilities as described in Section 4.3. The Removal Volumes generated become part of the spatial grammar's vocabulary and can be used by the rules as presented in Section 4.4. Using the fundamental knowledge of the machining process, material (i.e. volume) is removed from the TRV by iteratively applying rules of the spatial grammar as described in Section 4.5. To apply the rules in a directed manner, heuristic search methods are applied. Simultaneously, the application of rules instantiates the toolpath and the necessary CNC instructions to execute the operation on the machine tool. After each rule application, the TRV geometry, i.e. the working shape, is updated and further rules are applied until the volume of the TRV is minimized. After having minimized the volume of the TRV, the CNC instructions are collected from the created toolpaths in the sequence of rules applied. The resulting machining plan can be executed on a CNC machine tool.

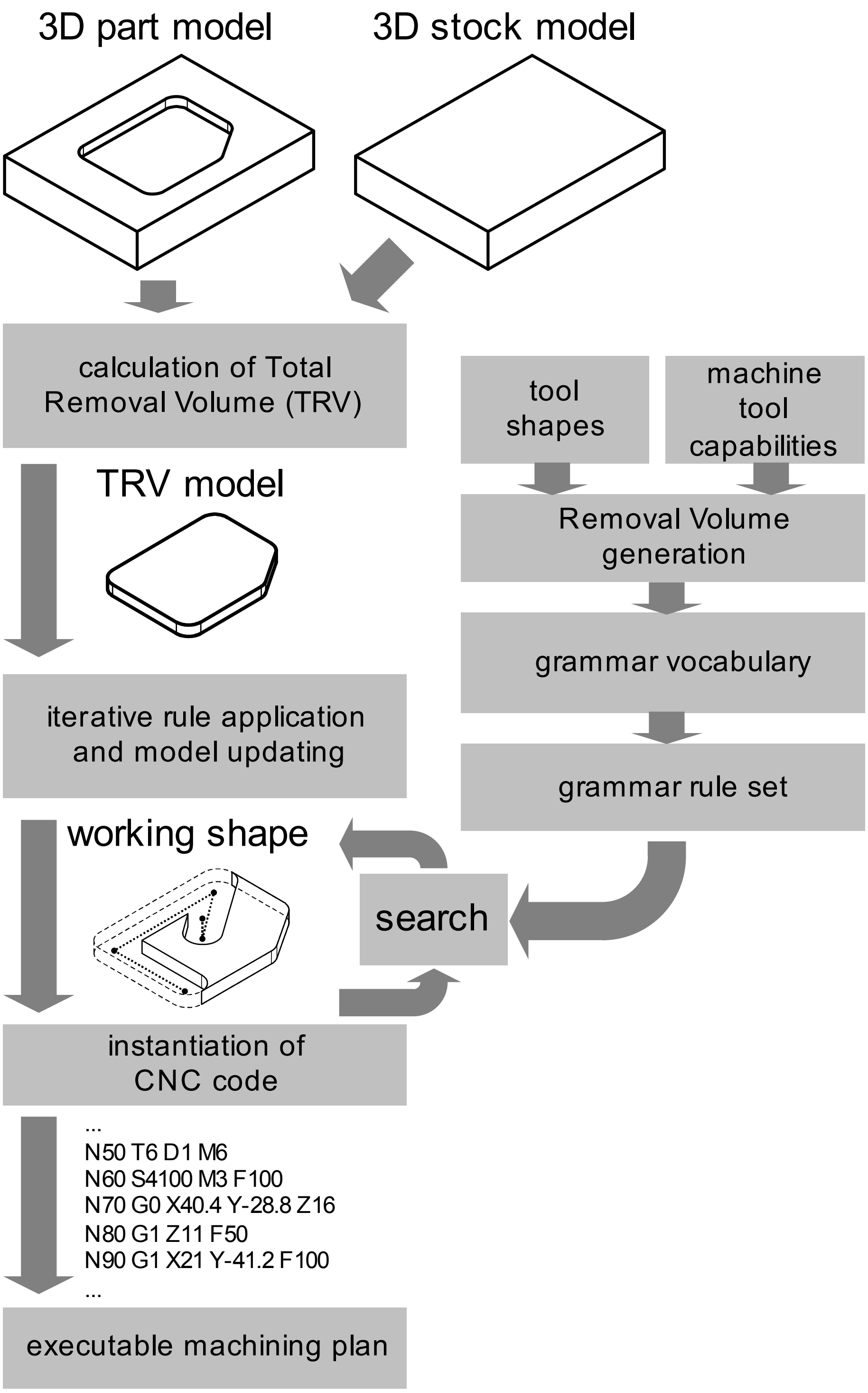


Figure 4-1: Design-to-fabrication process

4.2 Terms and definitions

This section presents basic terms and definitions that are used in the presentation of the method.

4.2.1 Total Removal Volume

The Total Removal Volume (TRV) represents the sum of all volumes that need to be removed from a stock (workpiece) to yield the desired part. As shown in Figure 4-2, the TRV (gray) is not necessarily connected but can consist of several bodies. In the expression of Boolean operations it is

$$TRV = stock - desired\ design \tag{1}$$

4.2.2 Open Face

Every face of the TRV that is shared with the stock but is not part of the desired part is called an Open Face (OF).

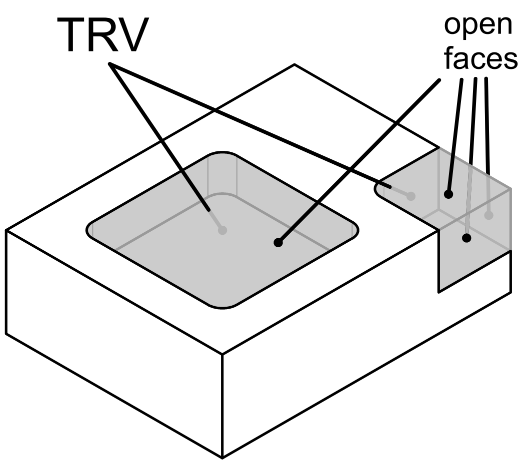


Figure 4-2: Total Removal Volume (TRV) of a part and its open faces

4.2.3 Removal Volume

The Removal Volume (RV) (Figure 4-3) represents the volume of material that can be removed from a stock by the interaction of the tool with the given stock during a single operation (movement of the tool) along the toolpath. The size of the RV is further constrained by the maximum cutting depth t. The shape of the RV depends also on the shape and size of the tool.

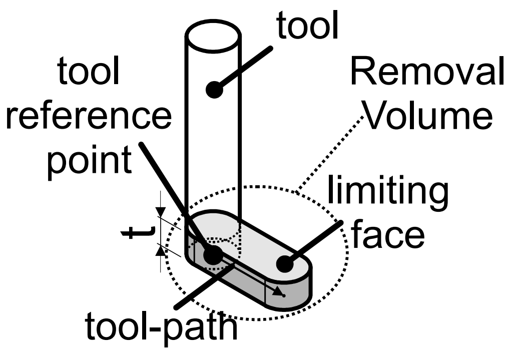


Figure 4-3: The Removal Volume (RV) created by a tool-sweep and constrained by the maximum cutting depth

The RV itself consists of the removed volume, the toolpath and constraints for the application of the RV and therefore contains all necessary information to execute the machining operation.

4.2.4 Limiting Face

The Limiting Face (LF) constrains the RV to a certain cutting depth t of the tool. The LF in most cases is the top-face of the RV as shown in Figure 4-3. The LF can be defined as the face created by offsetting the swept tool surface normal to the swept tool face and orthogonal to the toolpath towards the tool axis by the distance of the defined maximum cutting depth t. If tool-axis and toolpath are collinear, then the RV does not have a LF.

4.3 Generating removal volumes

To capture the knowledge of the machining process, a fitting representation is required. This representation has to have all the properties required for the application, i.e. the geometry-based generation of a process plan for machining on a CNC machine tool. This suggests using a geometry-based representation for the single operations that can be carried out on a machine tool. In this approach the representation comprises the Removal Volume (RV) that the machine tool and the tool can remove from the stock along with the toolpath, i.e. the instructions to carry out the operation.

In the area of machining simulation, approaches to generating the volume of a machining operation exist in the area of Cutter-Workpiece Interaction analysis (see ARAS & YIP-HOI 2008 for examples). An approach similar to ARAS & YIP-HOI (2008) based on MOUNAYRI et al. (1998) and SPENCE & ALTINTAS (1994), is used to generate the RV from a tool with a given elemental toolpath segment shown in Figure 4-4. The process comprises (a) decomposition of the solid tool representation into faces, (b) sweeping of the individual faces along the toolpath segment, (c) Boolean sum of the individual solids and (d) healing of irregular bodies (e.g. non-manifold and hollow bodies). This method is not bound to endmills, but can be used for a wide variety of milling tools (see ARAS & YIP-HOI 2008 for examples). In contrast to the initial approach presented in Section 3, the shapes of the RV depend on the tool shape. Through this, the approach can be applied using a wide variety of tool shapes.

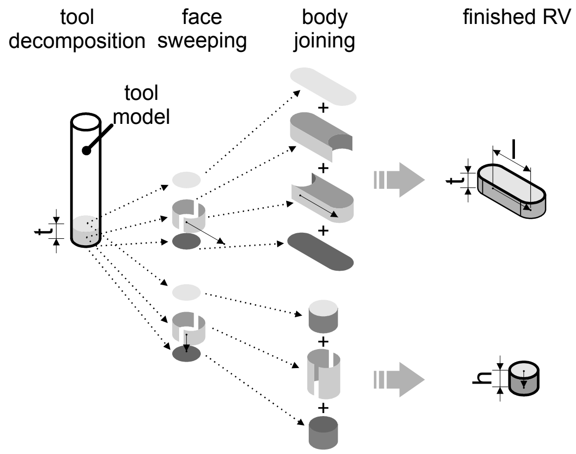


Figure 4-4: Removal Volume (RV) generation

The Removal Volumes also contain the parameters of the RV that are instantiated during rule application and the definition of the used machine tool, tool, toolpath and machining instructions as labels. The complete machining instructions can then be collected from a sequence of RVs.

The knowledge on the machining process is encoded in the rule set and in the vocabulary. In the vocabulary, the elemental shapes that can be removed by a machining operation are generated from the combination of the tool shape and the kinematic capabilities of the machine tool from the machine tool library.

4.4 Set grammar formalism

As described in Section 2.4, a set grammar consists of a vocabulary, the set of rules and an initial labeled shape. The vocabulary here, shown in Figure 4-5, consists of the Removal Volume shapes, toolpath-curves and endpoints of the toolpath. The set of labels (Figure 4-6) is used to control the application of rules and parametrically represent CNC commands.

The Total Removal Volume (TRV) represents the initial shape of the grammar. During the application of RVs to the TRV, the TRV and the sequence of RV applications represent the working shape.

In detail, the set of shapes consists of the vocabulary of Removal Volumes (Figure 4-5), which in turn consists of the solid shape of the removed volume by the machine tool, the toolpath of the associated machining operations, which are assumed to be straight lines and the start- and end-point of the toolpath.

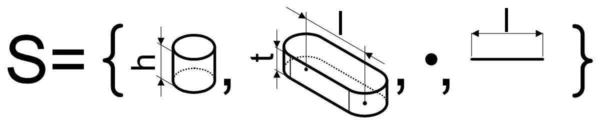


Figure 4-5: Basic vocabulary with Removal Volumes

The Removal Volumes (Figure 4-5) are constrained such that only the defined parameters, here *t*, *l* and *h*, are allowed to be changed to enable the generation of a feasible process plan.

The set of labels (Figure 4-6) consists of labels for Non-Open-Faces (NOF), Open Faces (OF), Limiting Faces (LF), Non-Limiting Faces (NLF) and a Marker (M) representing the current spatial position of the generation process thus ensuring a continuous toolpath. Further labels include G0 and G1 as CNC-commands for moving the tool at machining feed (*G1*) or rapid feed (*G0*) and the triplet *<x,y,z>* describing the coordinates of a point.

L= {NOF, OF, LF, NLF, M, G0, G1, <x,y,z>}

Figure 4-6: Set of labels

The set of rules, shown in Figure 4-7, describes how the vocabulary can be applied to the working shape. The rule set consists of the rule R.1 that applies the starting symbol on the model level, thus creating a toolpath origin on the machine level. The repositioning rule, R.2, creates a *G0* labeled line from the current marker position to a new position. On the machine level this represents a rapid, thus non-cutting, movement of the tool to a new position. The terminal rule, R.4, ends the generative process on the model level and retracts the tool from the workpiece on the machine level. The most important rule of the set is the subtractive set rule schema, R.3, for 3-axis-machining. This rule schema applies on the model level a Removal Volume under a certain transformation (rotation and definition of parameters of the RV) and thus removes the volume of the RV from the working shape while adding the toolpath to the working shape. Since the working shape can have an arbitrary shape, Figure 4-7 can only give an example of such an instantiated rule and major parameters. Newly created faces are labeled as Open Face (OF) unless they are coincident with a Non-Open Face (NOF). This corresponds to an actual single machining operation on the machine level, with the engagement of the moving tool and the workpiece.

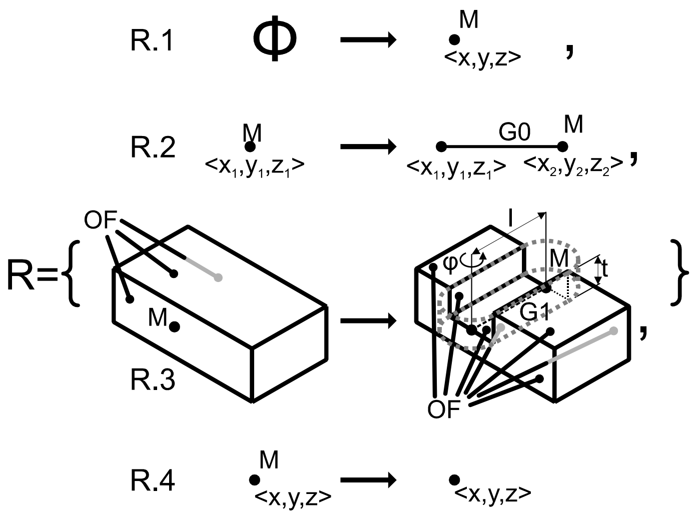


Figure 4-7: Set of rules

To allow for maintenance and extension, the grammatical system features a minimum number of rules to cover the fundamentally different operations in a manufacturing process. Such operations can be the cutting process and the movement associated with it but also repositioning of the tool at rapid feed.

In general, the application of a rule

$$A \rightarrow B \tag{2}$$

follows the formula

$$C' = C - T(A) + T(B) \tag{3}$$,

where *C'* is the new working shape generated from the former working shape *C* and the shape sets *A* and *B* under the transformation *T*. Specifically for rule R.3 (Figure 4-7) this schema results in

$$C' = C - T(V(RV)) + T(TP(RV)) \tag{4}$$

Where *V(RV)* represents the volume of the Removal Volume that is subtracted and *TP(RV)* represents the toolpath of the RV that is added to the working shape.

These rules are general referring to their use, but represent specific operations in the manufacturing process. Such operations can be the cutting process and the movement associated with it but also repositioning of the tool at rapid feed. As a drawback of this approach, the system must rely on numerous constraints on the spatial relation between the TRV and RV to ensure that only feasible and valid process plans are generated. The implementation and embedding of these constraints, however, is a known difficulty in spatial grammars. Encoding the constraints within the rules makes the constraints rule specific where some constraints may be repeated in several rules and would have to be changed in each rule. The definition of all constraints in the rules also requires creating a multitude of rules to cover all situations in which the explicitly defined constraints apply.

The hard constraints for the rule application are shown in Table 4-1. These may not be violated to ensure that a feasible plan is generated.

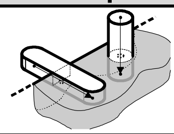
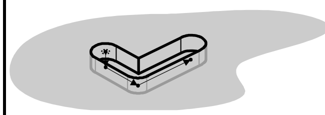
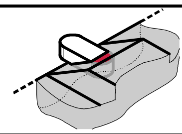
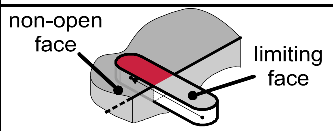
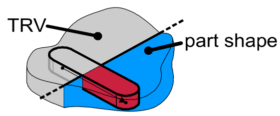
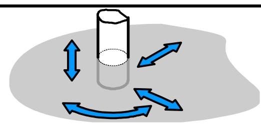
The constraint **C.1** resides within the rules R.1 and R.2. The toolpath must start outside of the Total Removal Volume thus preventing collisions of the tool and the TRV. This addresses also constraint **C.2**, which ensures a continuous toolpath, i.e. the tool cannot freely move in the material. This also reduces the computing complexity since it disallows translation during rule application such that the endpoint of the previous RV's toolpath becomes the starting point of the current RV's toolpath. In practice, this is ensured by the use of marker M from the set of labels in rule R.3.

Constraint **C.3** prevents the limiting face of the RV from penetrating the TRV. This prevents the tool from cutting too deep into the material resulting in exceeding the allowed cutting forces and possible tool-breakage.

Similarly, **C.4** forbids limiting faces of the RV to coincide with non-open face of the TRV. Referring to the definition of these terms, this constrains the tool from approaching the part from within the material.

Constraint **C.5** prevents the RV from penetrating the designed part shape, which would cause an irreversible deviation from the desired end-contour.

Table 4-1: Hard constraints on spatial relations for rule application

name	rule	spatial relation type	spatial relation	contribution to: plan feasibility	contribution to: reducing complexity	example
C.1	R.1 R.2	RV:TRV	The toolpath must start outside the TRV	X		
C.2	R.3	RV:RV	The toolpath must be continuous	X	X	
C.3	R.3	RV:TRV	A limiting face must not penetrate the TRV	X		
C.4	R.3	RV:TRV	Non-open faces must never be coincidental with the limiting face	X		
C.5	R.3	RV:Part	The part shape may not be penetrated by the RV	X		
C.6	R.2 R.3	RV	only translations x, y, z and rotations in dz are allowed	X	X	

C.6 constrains the possible transformation of the RV to translations in x, y and z directions and rotations around the z-axis, which is the case for 2.5D machining on a 3-axis milling machine. Once again, this is necessary to create an executable plan and also lowers the complexity by reducing the possible combinations of the transformations.

4.5 Heuristic search methods for machining planning

This section presents a new method for generating machining plans by combining the presented set grammar with heuristic search.

Many problems, especially in combinatorial optimization, require vast computational effort to enumerate all possible solutions to the problem posed. However, often the optimal solution is not required and instead, finding a near-best solution that is computable in reasonable time is acceptable. In such cases heuristics play an important role. Heuristics, or rules of thumb, are criteria and methods that help decide which alternatives to pursue further and which ones to abandon. A good heuristic balances the effort to compute it and its significance, i.e. its ability to discriminate correctly between a good and a bad choice.

Using the good guesses provided by the heuristic, the search problem can be constrained to promising areas. Through this, less partial solutions of a combinatorial problem must be evaluated, thus the computational effort is significantly reduced. However, using a heuristic does not guarantee finding a solution nor finding any solution at all. Summarizing, a heuristic can be defined as "… a technique that seeks good (i.e. near-optimal) solutions at a reasonable computational cost without being able to guarantee either feasibility or optimality, or even in many cases to state how close to optimality a particular feasible solution is" (REEVES 1995).

The search strategy determining which nodes of the search tree are generated or expanded next has a great influence on the algorithm's performance. Depending on the strategy, a search can be uninformed or informed (PEARL 1984): Uninformed search methods know little to nothing about the structure of the problem or the reachable states in the neighborhood of the current state. Examples are Depth-First Search (DFS) or Backtracking. In DFS, the search tree is expanded by picking the successor nodes at random. The search is terminated when the goal state is reached or single nodes are abandoned if the depth bound is reached or the node has no successors. If a node is a dead-end or has reached the depth bound, the search continues with another node. In contrast to DFS, backtracking generates only a single successor for each node and further pursues in this direction. If a node is a dead-end, the search returns to its predecessor and generates another successor, previously not visited. Through this, backtracking can recover from dead-ends. However, the search itself is still undirected. Informed search methods have the ability to assess the quality of an alternative state by means of a heuristic function. This function provides the search with information on how close the state is to the goal state. Examples for search methods that capitalize on this are Best-First Search (BFS) and A*. In the search the open nodes or states, i.e. the nodes that still need to be explored, are ranked by their respective value of the heuristic function. In BFS, the highest ranking successor node is explored. In contrast to BFS, A* uses not only heuristic information on how close the evaluated state is to the goal but also evaluates recursively the cost for this particular partial solution.

4.5.1 A* search

A* is a widely used search method for graph search due to its ability to use heuristic information in the search to guide it. A* is usually implemented using a ranked list of states to be explored, sorted by the heuristic function *f(i)* for each state *i*,

$$f(i) = (1-w)\,g(i) + w\,h(i)\,. \tag{5}$$

The cost is represented by the recursive function *g(i)* with

$$g(i) = g(i-1) + \Delta g(i) \tag{6}$$

and some weight

$$w \in [0..1]\,. \tag{7}$$

The term *g(i-1)* represents the previously generated cost, whereas *Δg(i)* is the cost caused exclusively by the state *i*. The function *h(i)* is an optimistic estimate of the distance to the goal, i.e. the quality of the state. Thus *h(i)* may never overestimate the distance and therefore needs to be admissible. If *h(i)* is monotonic, the search can be implemented more effectively because each state is only visited once.

During each iteration, the state/node with the lowest *f(i)* is selected and further expanded. By minimizing *f(i)* (eq. (5)), the goal can be reached effectively and with minimum cost. Since only the state with the minimum *f(i)* is pursued further within the search, the search can get trapped in local optima.

Figure 4-8 shows the tree of evaluated states using A* for a single depression. First, the starting symbol is applied, initializing the heuristic values *h*, *g* and *f*. From this state, three rules are applicable: ApplyRV with a cylindrical RV, ApplyRV with a slot RV and the Repositioning rule.

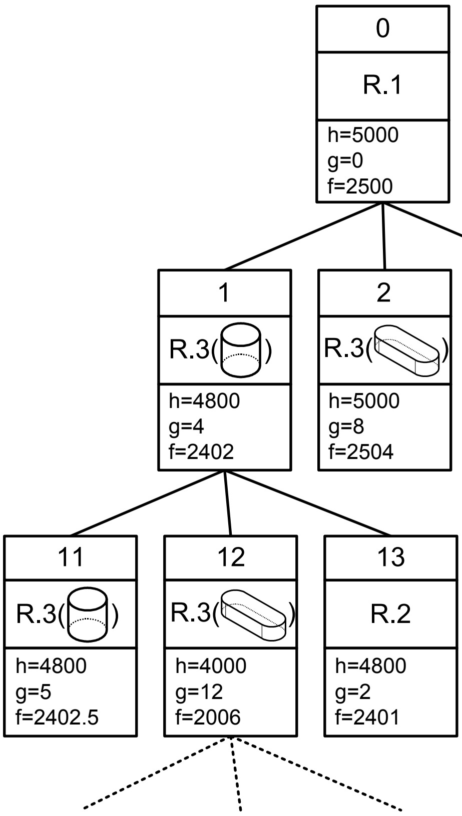


*Figure 4-8: Partial decision tree after the first few iterations using A**

A* evaluates all of these three rules, thus creating new states and calculating their objectives. After this, A* continues with the state having the lowest *f* value, here state 1, the only state that cuts away material and thus lowers the *h* value. From this, again the three applicable rules are applied and evaluated. However, applying a cylindrical RV fails since the marker is already at the bottom of the depression. Therefore, state 11 is not pursued further. State 12 has the overall lowest *f* value and is pursued further during the search. It can be seen that A* does not accept inferior moves that can have a positive influence later on search and thus cannot escape local minima.

4.5.2 Pattern Search

Pattern Search (PS) (HOOKE & JEEVES 1961) is known for its ability to solve geometric optimization and search problems well. Examples comprise 3D component layout problems (YIN & CAGAN 2000) and trunk packing (knapsack) problems (DING & CAGAN 2003).

In the Hooke and Jeeves Pattern Search, the generation of neighboring solutions is prescribed by alternately changing each variable, one at a time. If all variables have been attempted to change and if the solution obtained is better than the previous one, a so called pattern move can be performed. Having a vector $\vec{v}$ of variables at iteration *i*, the pattern move $\vec{m}$ can be calculated from the previous iteration *i-1* by

$$\vec{m} = \vec{v}_i - \vec{v}_{i-1} \,. \qquad (8)$$

Then, a change of the variables by repeating the movement pattern $\vec{m}$ from the last iteration can be applied with

$$\vec{v} = \vec{v}_i + \vec{m} \,. \qquad (9)$$

If the result of the objective function for the vector $\vec{v}$ is better, it is accepted. Otherwise, the search returns to the variables of the last iteration *i*. Through the use of pattern moves, the search can proceed according to an estimated gradient (see eq. (8)). As a result, PS can be applied even if a gradient cannot be calculated.

The PS search procedure is illustrated in Figure 4-9 where a 2D continuous minimization problem is considered. The curves represent the values of the objective function with the minimum in the center.

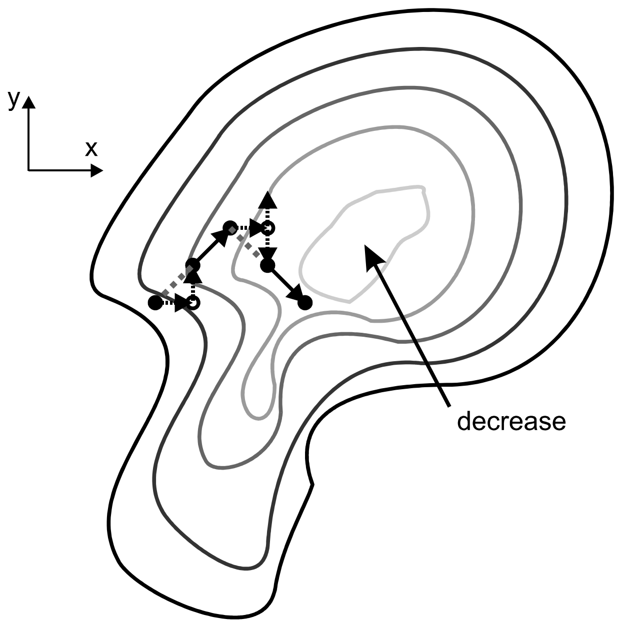


Figure 4-9: Example of Pattern Search

The first step is taken in the x-direction and accepted since the objective value is lower. The second move is attempted in the y-direction and also accepted. Once all variables have been changed, a pattern move according to eq. (8) is calculated and performed based on the current state (so called basepoint, shown as empty circle). The pattern move (displayed as a gray dotted line in Figure 4-9) is accepted since it decreases the goal value. The next move is again applied in the x-direction and accepted. The next move is attempted in the positive y-direction but rejected since the objective value is higher. Therefore, the same move in the negative y-direction is attempted and accepted since this time the objective value is lower. After all variables have been attempted to be changed, a pattern move is calculated again and accepted. The search continues until the minimum is reached (not shown).

4.5.3 Decomposition of the Total Removal Volume

The problem of applying a subtractive rule and thus decomposing the TRV into a sequence of RV applications can be sectioned into subproblems as follows:

a) (global search problem) find the best sequence of operations (rule applications) to yield the desired part

b) (local decision problem) select which removal volume to apply

c) (local search problem) find the transformation (position, orientation and parameters) under which the rule application is valid and most effective

The process of the decomposition is shown in Figure 4-10. The decomposition process is repeated until the volume of the Total Removal Volume is smaller than the a-priori defined Tolerance Volume. First a rule is selected. If the rule incorporates a Removal Volume, e.g. R.3, one parameterized Removal Volume from the set of all Removal Volumes is selected. At this point, the tool and machine tool on which the operation should be carried out is already determined through the inheritance of this information during the RV generation. The local search has then to find suitable transformations that allow the rule to purposefully be applied to the TRV under consideration of the constraints on the spatial relation. If all constraints are met, the rule can be applied to the working shape under the found transformation. After this, the process continues from the beginning. If there is no transformation that would allow the rule application, the process returns to a prior step and either selects a different RV or another rule to apply.

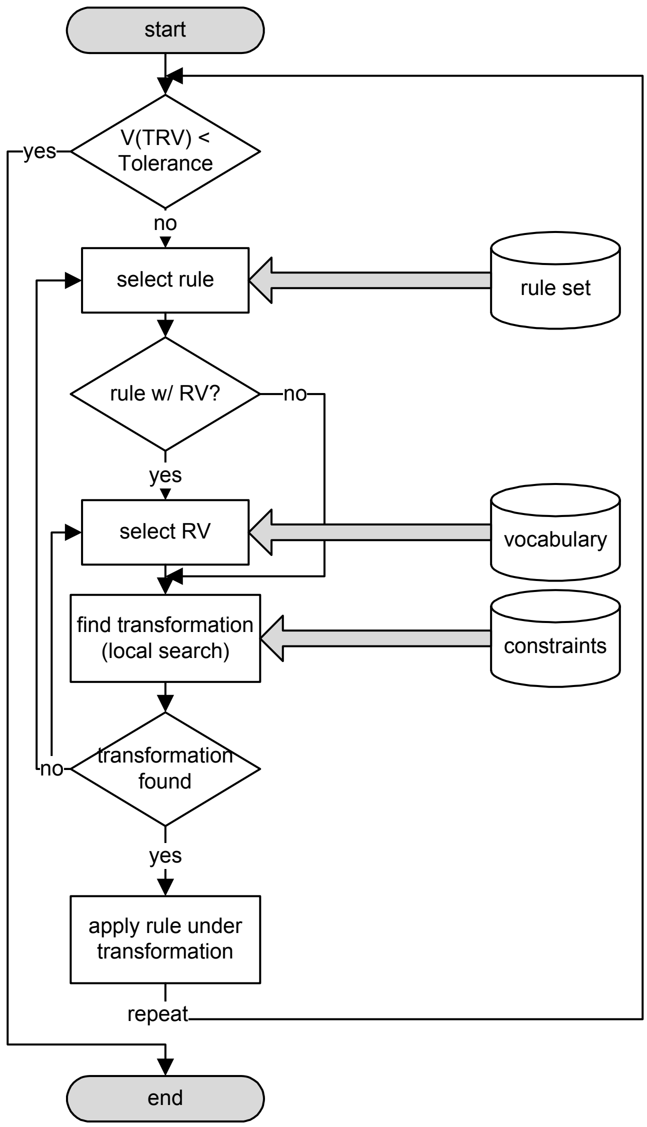


Figure 4-10: Total Removal Volume decomposition process

Figure 4-11 shows this decomposition process using an example. After the TRV has been calculated from the part model and stock model, the starting symbol (rule R.1) is applied to the TRV. The marker, M, is placed directly above the top surface of the TRV indicating the starting point for cutting. Then rule R.3 is instantiated with the cylindrical Removal Volume along with parameter h, and applied, representing a vertical tool movement. The toolpath is created at the same time, denoted by the dotted line, and the position of the marker M is moved to the endpoint of the toolpath. Next, the same rule is instantiated and applied again, this time using the bar-like RV and parameter l, representing a horizontal tool movement. The toolpath is generated, denoted by the dotted line. The application of rule R.3 using the bar-like RV continues until the TRV has been completely removed. Then, the terminal rule R.4 is applied, thus retracting the tool to a safe position and removing the marker *M*. During all rule applications the CNC instructions are instantiated such that the CNC control program to drive the machine tool can be collected from the sequence of successful rule applications.

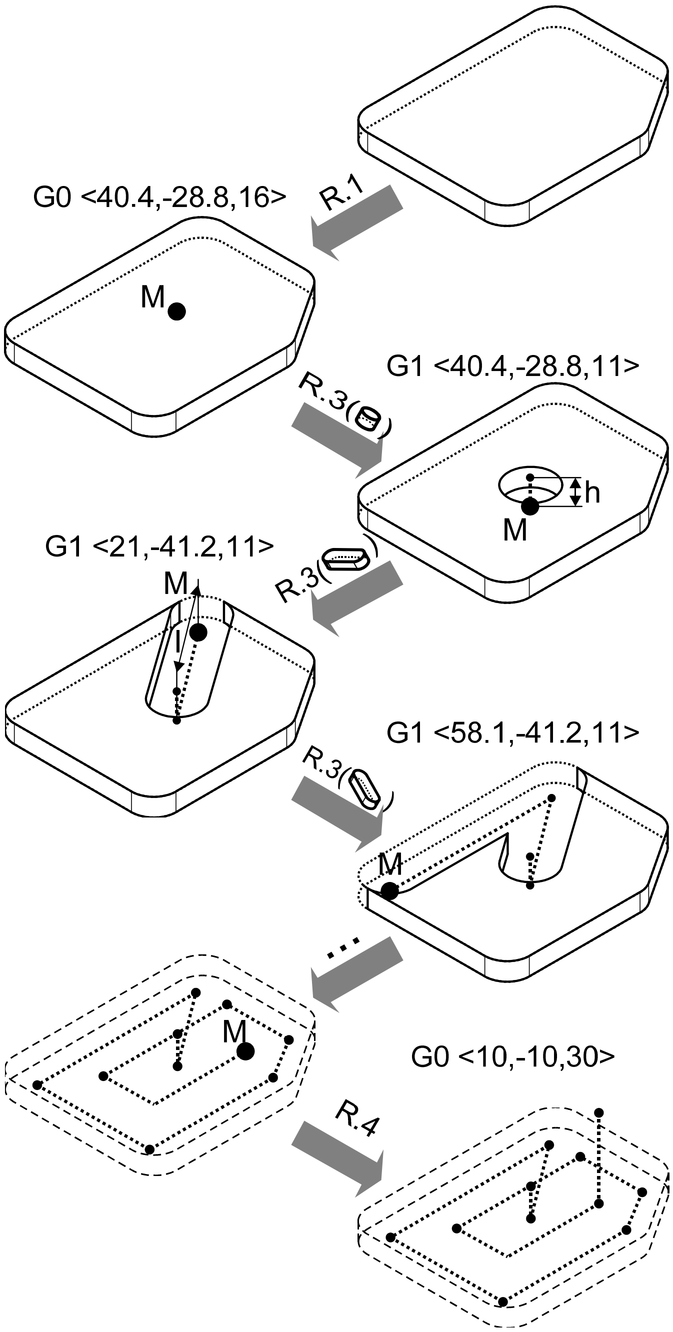


Figure 4-11: Sequence of rules for the decomposition of a TRV

The search problem for this application can be split into two loops:

- An outer loop to select which rule to apply and to search the tree of dynamically generated states each formed by a rule application, addressing subproblems a) and b).
- An inner loop that optimizes continuous parameters for the Removal Volumes that are subject to several constraints, addressing subproblem c).

Due to the search algorithms' requirements, a more complex objective function for minimization is used instead of only minimizing the volume of the residual TRV as depicted in Figure 4-10.

Outer loop search

For the outer search, i.e. the search for a good sequence of rule applications, the A* algorithm (PEARL 1984) has been chosen for its ability to use heuristic information on benefit and cost of actions to guide the search through all generated states.

Assuming *i* denotes a state, the goal distance function *h*, represents the benefit of each action and is chosen as the residual volume of the TRV, i.e. the remaining volume to remove:

$$h(i) = Volume(TRV(i)) \qquad (10).$$

The *h(i)* heuristic must be admissible i.e. it must not overestimate the distance to the goal. Since the Volume of the TRV should be made equal to zero, *h(i)* is exactly the distance to the goal and therefore admissible.

The cost *g(i)* of each action or rule is represented by the approximated time it takes to fabricate a part up to this state, thus it can be recursively defined

$$g(i) = time(i) + g(i-1) \qquad (11),$$

where *g(i-1)* refers to the cost of the previous state. The A* algorithm combines the metric for benefit and cost into one function *f* using the weight $w \in [0..1]$ to balance the guidance between benefit (i.e. greediness) and cost according to the formula:

$$f(i) = w \cdot h(i) + (1-w) \cdot g(i) \qquad (12).$$

The goal is to minimize *f(i)* during the outer loop search and thus find a feasible machining plan with minimum cost and minimum deviation from the desired part shape.

In addition to the cost function *g(i)* and the goal distance function *h(i)*, a penalty is added to allow relaxation of constraints. In analogy to (12), a weighted objective function can be formulated as

$$f(i) = w \cdot h(i) + (1-w) \cdot g(i) + w_p \cdot (p(i)^2 + p_{offset}) \qquad (13),$$

using the weight $w_p \in [0..1]$ to control the influence of the penalties in comparison to cost and benefit with the penalty value

$$p(i) = p(i-1) + \Delta p(i) \tag{14}$$

and a positive penalty offset p_{offset} defined as

$$\begin{aligned} p_{offset} &\in [0..\infty); \quad p(i) > 0 \\ p_{offset} &= 0; \quad p(i) = 0 \end{aligned} \qquad (15).$$

The goal of the search is to minimize the objective function value *f(i)* according to eq. (15).

The A* search algorithm, as implemented, is outlined below:

Table 4-2: Basic A algorithm in pseudo code for searching a tree of states*

```
create root of state tree and put node on open list
while openlist (sorted by descending f(i)) is not empty
    select node with the lowest f value and make it the current state
    if current state meets goal criterion
        exit search
    remove the state from the open list
    generate all state siblings
    calculate g, h, p and f for each new node
    add each node to the open list
    add current state to the closed list
```

Inner loop search (parameters of Removal Volumes)

The inner loop search is used to optimize the parameters of each candidate Removal Volume. For this problem, Pattern Search (PS) has been selected for its known good behavior on geometric optimization problems (YIN & CAGAN 2000). Since the outer loop A* search requires the use of an objective function, incorporating cost and benefit, it is reasonable to use the same objective function in the local loop of the search process as well.

However, since Pattern Search explores the search space deterministically, it can get trapped in certain solutions, thus returning no result. In addition to this, Pattern Search can also fail to find a feasible solution under hard constraints. The cause of this behavior is discussed in the following.

Figure 4-12 presents a situation where the deterministic search space exploration can block the search from proceeding under the hard constraint C.5. This constraint, shown in Table 4-1, forbids the RV to intersect the design part shape. The methods using hard constrained Pattern Search (a) and the penalized Pattern Search (b) are compared.

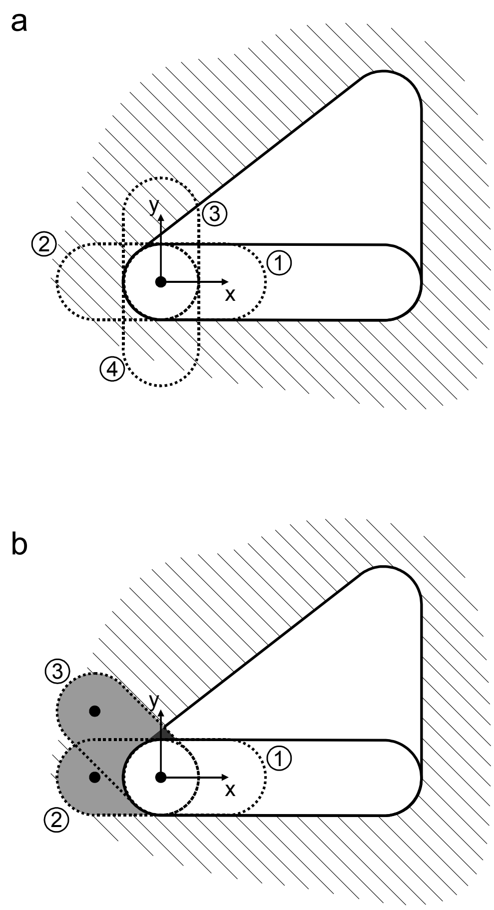


Figure 4-12: Hard constrained (a) vs. penalized (b) Pattern Search

The shaded area represents the part, whereas the inner clear triangle represents the valid machining area. The tool has already removed some of the volume from right to left in the lower area of the triangle. The marker, representing the current position of the virtual tool, is located at the origin of the coordinate system. In (a), the hard constrained Pattern Search is shown. First, the PS attempts a move in positive x direction (1). For this move however, the objective function increases since the move only creates cost *g(i)* while not changing the residual volume *h(i)* because the overlap of the RV and the TRV is zero. Therefore the move is not accepted. For the other explored moves (2-4), the constraint C.5 is violated since the RV (dashed outlines) intersects with the part shape. The penalized PS (b) does not accept the first attempted move (1) as well but is able to accept move 2 and 3 sequentially since these no longer violate the constraints and do have an effect to improve the objective function. The solid gray areas represent the volume that is added as a penalty to the objective function, whereas the small black area represents the volume that could be removed from the stock and which lowers the objective function.

If the hard constraint C.5 (intersections between Removal Volumes and the part shape are not allowed) is relaxed and transformed into a penalty which can be formulated recursively as the volume of the intersection between the Removal Volume that is applied and the part shape and the penalty of the parent state, thus:

$$p(i) = Volume(RV(i) \cap Part) + p(i-1) \quad (16).$$

In case there is no Removal Volume involved, the formula is simplified to.

$$p(i) = p(i-1) \quad (17).$$

The positive penalty offset p_{offset} is defined as

$$\begin{aligned} p_{offset} &\in (0..\infty); \quad p(i) > 0 \\ p_{offset} &= 0; \quad p(i) = 0 \end{aligned} \quad (18).$$

The penalty is used in eq. (9) to calculate the objective function. However, through early studies, it was found that the inner loop search (here Pattern Search) must not take cost into account otherwise it is possible that an invariant partial solution is returned, i.e. the algorithm chooses minimizing the cost *g(i)* over minimizing the residual volume *h(i)*, resulting in no change in *h(i)*. This keeps the outer loop search from achieving any progress through the search space at all. Therefore, the objective function for the local search loop is defined as

$$f_{local}(i) = w \cdot h(i) + w_p \cdot (p(i)^2 + p_{offset}) \quad (19).$$

The goal of the search is to minimize the objective function value *f(i)* according to eq. (19).

The Hooke and Jeeves Pattern Search algorithm, as implemented, is outlined below:

Table 4-3: Hooke and Jeeves Pattern Search algorithm in pseudo code

```
do
    for each variable x of x ∈ v⃗ in turn
        x = x + step_size(x)
        if objective value C_i=f(v⃗) decreased
            accept move and continue
        else
            revert move
            x = x - step_size (x)
            if objective value C_i decreased
                accept move and continue
            else
                revert move
    end of for loop
        if no move has been made
            reduce step sizes
        else
            attempt a pattern move v⃗ = v⃗ + m⃗
            if objective value C_i=f(v⃗) decreased
                accept move
            else
                    reject move
while goal criterion has not been reached and step_sizes not below minimum step_sizes
```

4.6 Implementation

The method for Spatial Grammar Machining Planning (SGMP), as presented, has been implemented using the C++ language and OpenCascade (OPEN CASCADE S.A.S. 2011) as the geometric kernel and for exchanging STEP-geometries, but not feature models, for parts and workpieces.

To allow for a high degree of flexibility of the grammar, it has been implemented using an abstract model of the grammar as shown in Figure 4-13. The main element is the abstract class SGMP for Spatial Grammar Machining Planning. It provides routines to instantiate Rules, Removal Volumes and Constraints using a factory pattern, implemented in the Loki library (ALEXANDRESCU 2001) during runtime. It also provides the functionality for searching the best parameters, and at the same time transformations, for each RV. The Total Removal Volume, the Non-Open Faces and the Part Shape are represented by the OpenCascade boundary representation class TopoDS_Shape. For the global search problem, the SGMP provides a tree structure (PEETERS 2011) of states to build up dynamically a search tree in which each rule application corresponds to a node. Each State saves the rule and the associated Removal Volume along with the Total Removal Volume. Through this, the search can return to previous states. Each Rule can be applied to the current state after the associated constraints have been checked. If a rule features a Removal Volume, the constraints for the Removal Volume are checked as well.

During the local search, the Parameter Vector, representing the shape and transformation of the RV, is modified and the shape and the Limiting Face of the Removal Volume are generated. Then the constraints can be evaluated. If Rule and Removal Volume have been successfully applied, the CNC instructions can be instantiated.

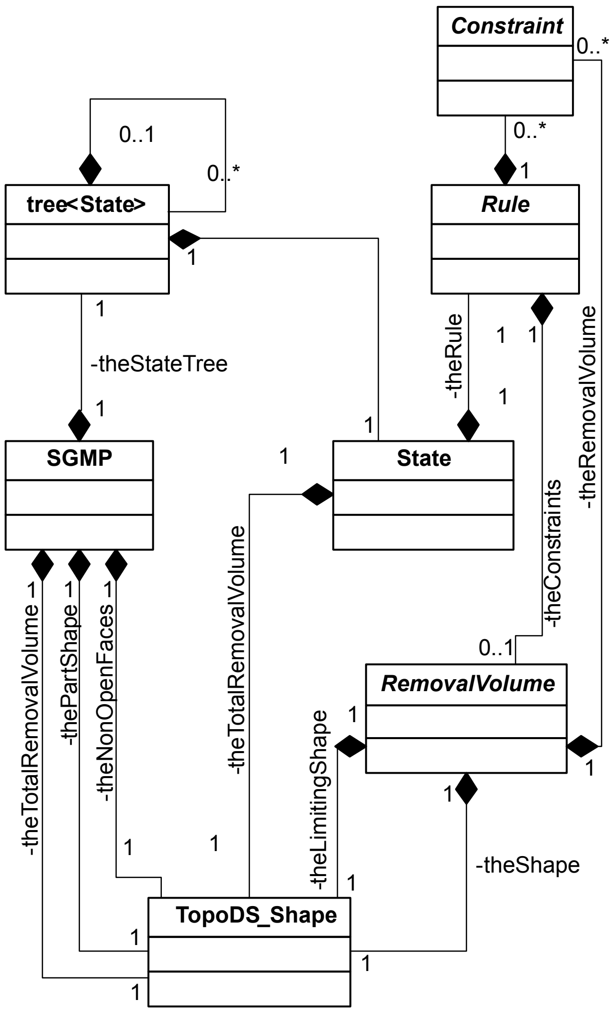


Figure 4-13: Basic UML diagram of the abstract grammar implementation

This structure provides the framework for the grammar and cannot be compiled as the vital functionality must be defined in derived classes. To create a working grammar, rules, Removal Volumes and constraints must be derived from the abstract class definitions and their functionality realized by overwriting the abstract routines from the base class. Due to the design of the C++ language, each element must be registered in the constructor of the grammar class derived from SGMP. However, it is still possible to enable and disable elements of the grammar, such as rules or removal volumes, and change mappings of the constraints during runtime.

The grammar has been implemented using the structure described. The grammar of the implemented software prototype features the following rules, Removal Volumes and constraints (see Table 4-1) shown in Figure 4-14. The Starting Symbol (R.1 in Figure 4-7) is not associated with a constraint since it is only applied once during the search procedure. The rule Reset Marker (R.2) repositions the marker to a new position and therefore depends on the Marker to be already set that is expressed by the associated constraint. The Terminal rule (R.4) can also be only applied if the marker has already been set. The rule Apply Removal Volume is associated with two Removal Volumes representing a vertical and a horizontal tool movement (see Figure 4-5). The rule itself relies on an existing marker. The Removal Volumes associated are subject to the constraints, shown in Table 4-1, C.3 (The Limiting Face of the RV may not intersect with the TRV), C.4 (the Limiting Face and the Non-Open Faces may not intersect) and C.5 (the part shape and the RV may not intersect).

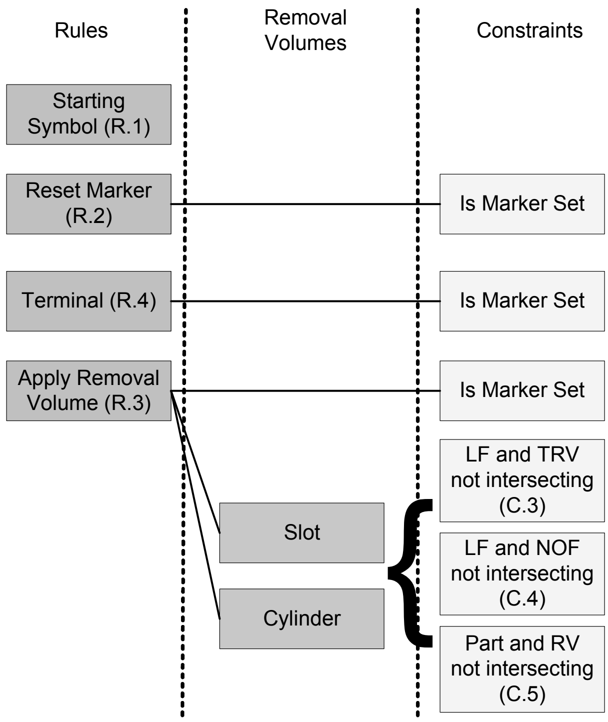


Figure 4-14: Implemented Rules, Removal Volumes and Constraints

Due to the nature of C++, the implemented elements of the grammar need to be registered in the factory pattern in order to be available in the grammar. The code in Table 4-4, taken from the constructor of the grammar's main class, shows the registration of the basic rules, Removal Volumes and Constraints. In the registration, a symbolic name (e.g. "ApplyRV") is defined, under which the element is accessible within the grammar. Along with the symbolic name, a pointer to a class (e.g. "RApplyRV) is specified that instantiates the object. Here, a template function is used to instantiate the object, so it does not require implementing a new function for each element manually.

Table 4-4: Code sequence for registering elements of the grammar

```
RegisterRule("StartingSymbol", createInstance<Rule, RStartingSymbol>);
RegisterRule("ResetMarker", createInstance<Rule, RResetMarker>);
RegisterRule("ApplyRV", createInstance<Rule, RApplyRV>);
RegisterRule("Terminal", createInstance<Rule, RTerminal>);

RegisterRemovalVolume("CylinderD10", createInstance<RemovalVolume, RVcylD10>);
RegisterRemovalVolume("SlotD10", createInstance<RemovalVolume, RVslotD10>);

RegisterConstraint("IsMarkerSet", createInstance<Constraint, CIsMarkerSet>);
RegisterConstraint("LFTRVnonintersecting", createInstance<Constraint, CLFTRVnonintersec>);
RegisterConstraint("NOLFnoncoincidential", createInstance<Constraint, CNOFLFnoncoin>);
RegisterConstraint("NOFRVnonintersecting", createInstance<Constraint, CNOFRVnoninter>);
RegisterConstraint("NoRepRules", createInstance<Constraint, CNoRepRules>);
RegisterConstraint("SameTool", createInstance<Constraint, CSameTool>);
```

Once the elements are registered, mappings that define the relations between the different elements are created. These mappings are checked during the instantiation of the elements such that related elements are created and added to the original element, e.g. once a rule is created, all associated constraints are also instantiated and added to the rule. The code in Table 4-5 shows the mapping for the grammar structure depicted in Figure 4-14.

Table 4-5: Code sequence for defining mappings between elements of the grammar

```
AddRuleToConstraintMapping("ResetMarker", "IsMarkerSet");
AddRuleToConstraintMapping("ApplyRV", "IsMarkerSet");
AddRuleToConstraintMapping("Terminal", "IsMarkerSet");

AddRemovalVolumeToConstraintMapping("CylinderD10", "NOFRVnonintersecting");
AddRemovalVolumeToConstraintMapping("SlotD10", "NOFRVnonintersecting");

AddRemovalVolumeToRuleMapping("CylinderD10", "ApplyRV");
AddRemovalVolumeToRuleMapping("SlotD10", "ApplyRV");
```

The first block defines constraints as being required by the specific rule. The second block defines which constraint has to be kept by which Removal Volume. The last block defines which Removal Volumes can be used by which grammar rule.

The rules, Removal Volumes and constraints can be deactivated or activated globally by unregistering or registering the element of the grammar. This illustrates the flexibility of the implementation.

5 Results and applications of Spatial Grammar Machining Planning

This chapter presents the results and several scenarios that demonstrate the potential of Spatial Grammar Machining Planning (SGMP) method as presented in the previous section. First, the feasibility of the approach and its implementation is presented, followed by a demonstration of the systems capability to autonomously plan the machining operations for the fabrication of customized parts. Then the handling of several disjoint bodies within one Total Removal Volume and the advantageous use of several tools is considered. After this, the system's ability to react to missing workpieces and the reaction to tool failures is demonstrated.

These scenarios cover a range of practical applications and address several common issues in machining planning as well as demonstrate the capabilities and flexibility of the method and its implementation. The machining plans have been created using the implementation and search methods, namely A* for rule selection and penalized Pattern Search (PS) for parametric optimization, as presented in Section 4.

5.1 Machining planning for part fabrication

The approach and implementation have been tested on several cases. The general behavior of the search is presented using the part shown in Figure 5-1. Here, the wedged depression must be machined from an oblong shaped stock.

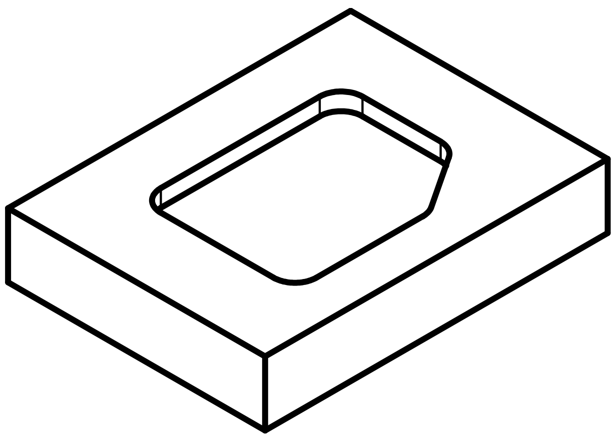


Figure 5-1: Basic validation part for search methods

First, the effectiveness of the pattern search is investigated for a single search for variables of the Removal Volume shown in Figure 5-2. The search first applied a cylindrical RV using rule R.3 based on the previous Starting Symbol that located the marker above the centroid of the pocket. Therefore, the marker is currently at the center bottom of the pocket, displayed as a cross in the figure. The pattern search must now find the best new location for the marker on an x-y-plane thus creating a toolpath from which the machine tool instructions can be created. From the current marker position, the search applies the first move in the positive x-direction. The search generates the Removal Volume and evaluates its volume of intersection with the TRV. The search continues applying the next move in the positive y-direction. Now that all variables have been attempted to be changed, a pattern move can be performed. For this, the current position of the marker is set as basepoint. The vector from previous iteration of pattern search to the current basepoint (eq. (8)) is added to the markers position (eq. (9)). The pattern move is indicated in Figure 5-2 as a diagonal, gray dotted line.

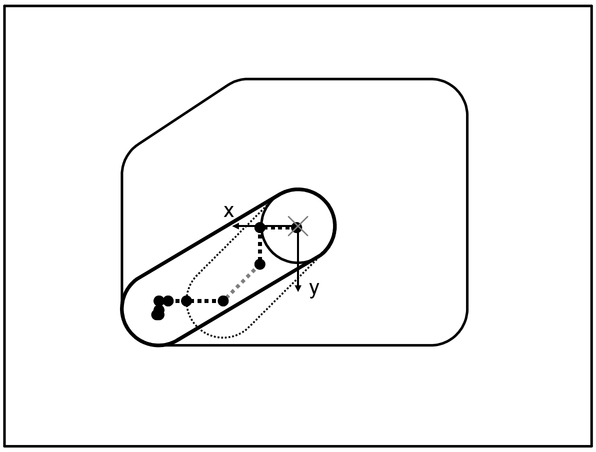


Figure 5-2: Pattern search procedure for a single Removal Volume. Pattern moves displayed diagonally in gray.

Figure 5-3 shows the development of the objective function over the iterations during the search. The objective function decreases steeply over the pattern move after the second iteration. After 18 iterations the step length is less than 0.001mm and the search procedure is terminated.

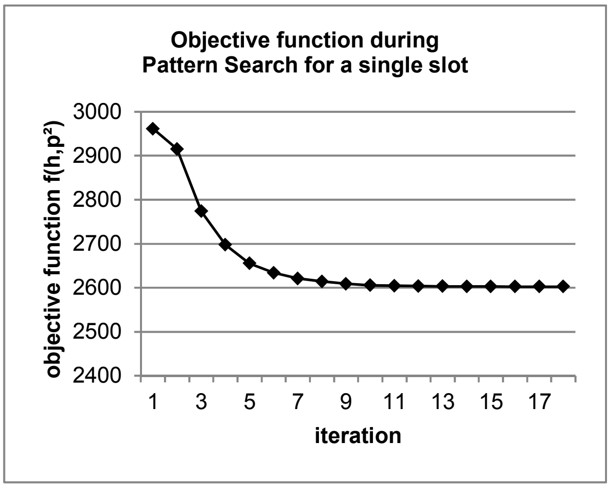
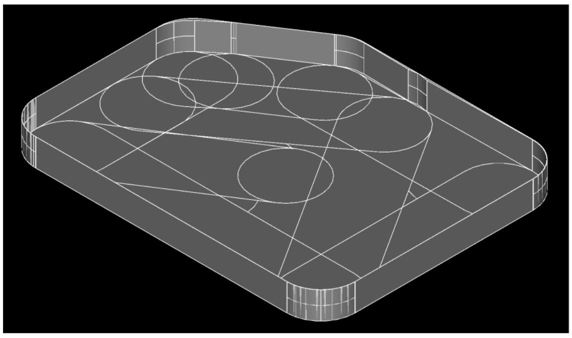


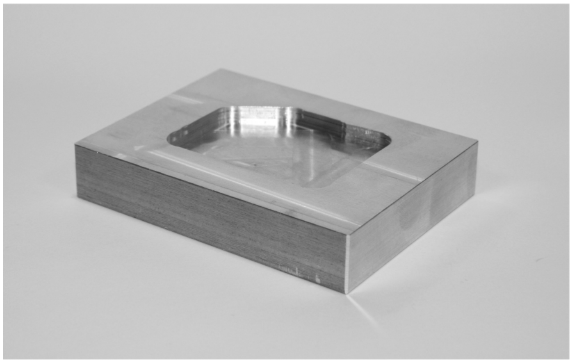
Figure 5-3: Objective function and iteration of the pattern search during the search shown in Figure 5-2.

To investigate the interplay of inner and outer search, the machining planning has been carried out repeatedly using a weight of $w=0.5$ and $w_p=1.0$ (cp eq. (13) & (19)). The resulting residual TRV volume $h(x)$ and cost $g(x)$ for each solution x have been sampled. Figure 5-4 shows one possible process plan for manufacturing the pocket (a) and a resulting part after machining (b) using CNC code generated during the planning process.

The search is carried out using a combination of A* and PS with hard constraints (in the following called method M1) and using A* and PS with penalties (in the following called method M2) as presented in the previous section. The objective functions are calculated for the outer loop (A*) with eq. (12) and for the inner loop (PS) with eq. (19).



a



b

Figure 5-4: Geometric representation of a generated process plan (a, colors enhanced for clarity) and a fabricated part (b)

Figure 5-5 displays the approximated cost vs. the normalized residual volume of the TRV after the terminal rule has been applied and thus the planning process has ended. The ideal solution lies in the lower left corner, with zero residual volume *h(x)* and minimum cost *g(x)*.

The diamonds represent solutions for the first search method (M1) using hard constraints instead of penalties. The narrow scattered datapoints indicate the general feasibility of the method.

The triangles represent solutions for the second search method (M2) using penalties instead of the hard constraint. These solutions can be grouped into three clusters by visual inspection. The first central cluster (cluster 1) is in the vicinity of the previously presented solutions. The second cluster (cluster 2) in the upper left corner represents solutions that are far from the ideal solutions, i.e. the method is not able to find acceptable solutions in these cases. The only difference between the methods is the different implementation of the constraints. Therefore, it can be assumed that the different behavior results only from this.

The third cluster (cluster 3) is the solution at the bottom that represents the overall best found solution. This solution demonstrates the potential of method M2 for finding improved solutions.

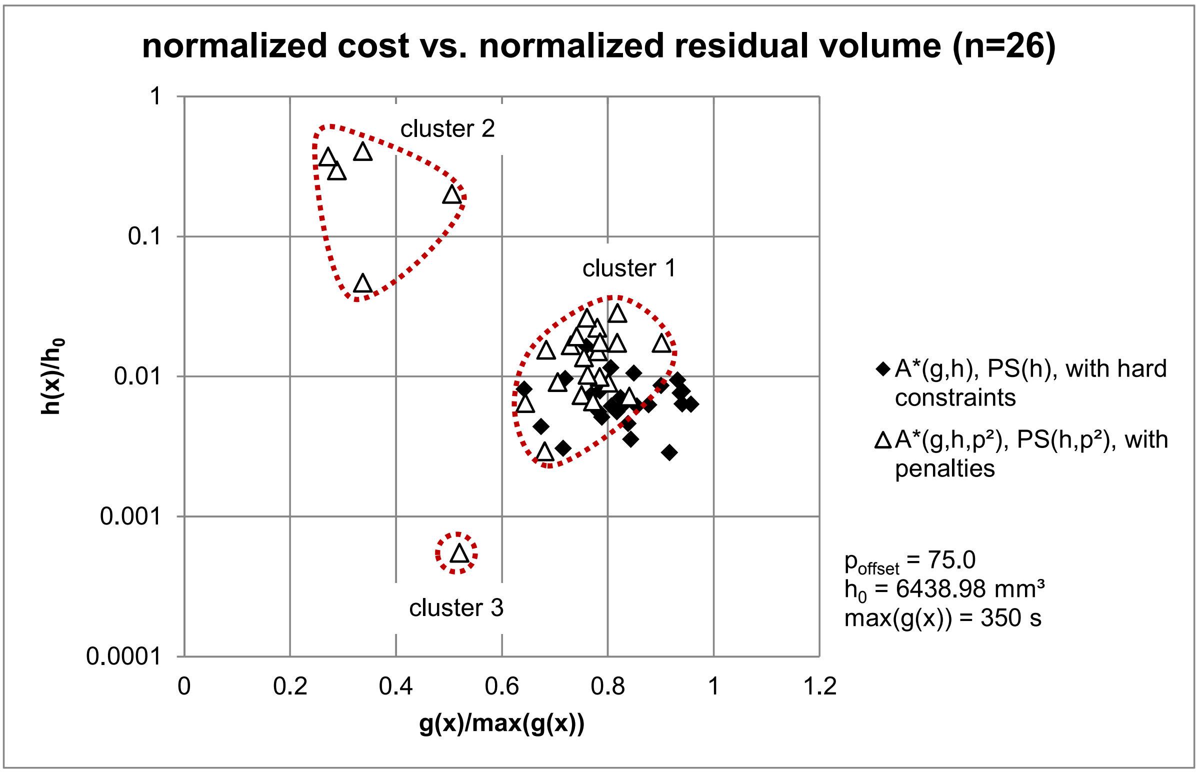


Figure 5-5: approximated cost vs. normalized residual volume (error value) for A-PS with hard constraints (M1) and A*-PS with penalties (M2).*

Ideally the residual volume *h(x)* would be zero for each solution *x*. The fact that it is not implies numerical problems or deadlocks in the search process. The numerical problems can be caused in the local search algorithm itself or the underlying CAD kernel and its Boolean functions for solids. The deadlocks of the search algorithm, i.e. it is not able to recover from a bad sequence of operations and thus is unable to reach the goal state, must be addressed by more powerful heuristics and guidance in the search process.

Table 5-1 shows detailed result statistics for each method. The first column shows the best solution for *h*, i.e. the solution with the least residual volume thus approximating the desired part best. The second column shows the best solution considering the combined objective function, *f,* as defined in eq. (12) and (19). The third and fourth columns show the worst solutions regarding *h* and *f* respectively. The fifth column shows the average values over all solutions and the last column the standard deviation over all solutions. Overall, the penalty *p* is zero for all solutions where it applies. This means, an unwanted overcut, i.e. removing material from the later part, did not occur in any solution. This demonstrates the ability of the method to find feasible solutions using penalties.

Table 5-1: Result statistics by methods, best and worst solutions for h and f, average values and standard deviations

		best min(h)	best min(f)	worst max(h)	worst max(f)	average	standard deviation
method M1	h [mm³]	18.4227	28.2290	105.6870	60.7200	45.6627	18.2203
	h_{res} / h_0	0.0029	0.0044	0.0164	0.0094	0.0071	0.0028
	g [s]	320.7200	235.9030	265.9890	326.2240	289.5924	29.0383
	p [mm³]	-	-	-	-	-	-
	f	169.5714	132.0660	185.8380	193.4720	167.6275	16.7012
method M2 overall	h [mm³]	3.5356	3.5356	2614.4600	2614.4600	395.3728	734.2194
	h_{res} / h_0	0.0005	0.0005	0.4060	0.4060	0.0614	0.1140
	g [s]	181.9350	181.9350	118.1500	118.1500	236.3794	62.1644
	p [mm³]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	f	92.7351	92.7351	1366.3100	1366.3100	315.8762	342.6549
method M2 cluster 1	h [mm³]	18.7894	18.7894	182.8270	182.8270	89.4167	43.1980
	h_{res} / h_0	0.0029	0.0029	0.0284	0.0284	0.0139	0.0067
	g [s]	238.2840	238.2840	286.4230	286.4230	267.7135	20.2066
	p [mm³]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	f	128.5370	128.5370	234.6250	234.6250	178.5650	26.7904
method M2 cluster 2	h [mm³]	299.3140	299.3140	2614.4600	2614.4600	1697.5648	833.6033
	h_{res} / h_0	0.0465	0.0465	0.4060	0.4060	0.2636	0.1295
	g [s]	118.1760	118.1760	118.1500	118.1500	121.9321	29.0411
	p [mm³]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	f	208.7450	208.7450	1366.3100	1366.3100	909.7492	411.8404
method M2 cluster 3	h [mm³]	3.5356	-	-	-	-	-
	h_{res} / h_0	0.0005					
	g [s]	181.9350					
	p [mm³]	0.0000					
	f	92.7351					

It can be concluded that method M2 is, overall, not better than method M1. Although method M2 is able to find an absolute best solution, the average values and standard deviations are larger than those of method M1. However, breaking the statistics down to the clusters identified in Figure 5-5, more conclusions can be drawn. Cluster 1 of method M2 shows similar results as method M1. The best solutions (*min(h)*) are almost identical in terms of *h*. The average values and standard deviation are larger than those of method M1. For method M2 the best solutions with *min(h)* is identical to the *min(f)* solution. Together with the lower spread and average values of the cost *g*, this indicates a more directed search towards effective solutions. Cluster 2 of method M2 displays overall unacceptable solutions with high average values and high standard deviations. Cluster 3 of method M2 is the absolute best found solution in terms of minimum *f*, *h* and *g*.

There are several reasons for the deviation of the results from the ideal goal state. First, the spatial resolution of the boundary representation is limited. Second, the minimum step length, i.e. the resolutions, of the heuristic search is limited. Third, the volume calculation needed for the objective function can cause numerical inaccuracies, especially for thin bodies. Finally, the search process can be trapped in local optima such that the residual TRV cannot be further removed.

The behavior of the two presented search methods is explored, using a slightly more complex example, depicted in Figure 5-6. Here the L-shaped depression must be machined from an oblong shaped stock.

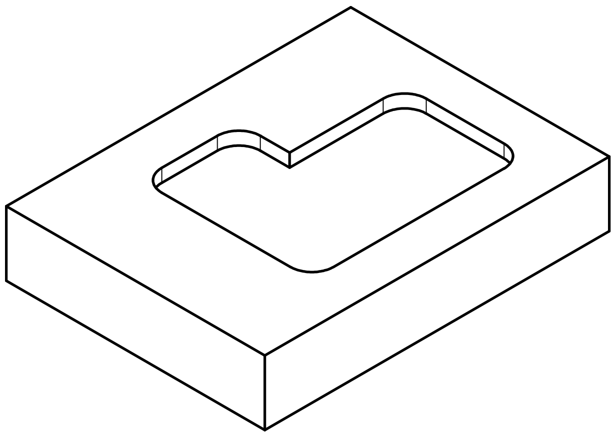


Figure 5-6: More complex example with an L-shaped depression.

The comparison of the two presented methods is shown in Figure 5-7. Again, the diagram shows the normalized residual volume and the approximated cost for machining the part with the ideal solution in the lower left corner. The diamonds represent method M1 whereas the triangles represent method M2. The results using method M1 form a loosely scattered pattern with a normalized residual volume above 0.0144 and normalized cost above 0.685. In contrast to this, the results using method M2 can be separated into three clusters. The second cluster (cluster 2) in the upper left corner represents solutions that are far from the ideal solutions, i.e. the method is not able to find acceptable solutions in these cases. The results within cluster 1 are more closely scattered than those of method M1, below a normalized residual volume of 0.0411. These solutions have a lower normalized residual volume with similar normalized cost. Cluster 3, consisting of only one solution is much closer to the goal state than any other result.

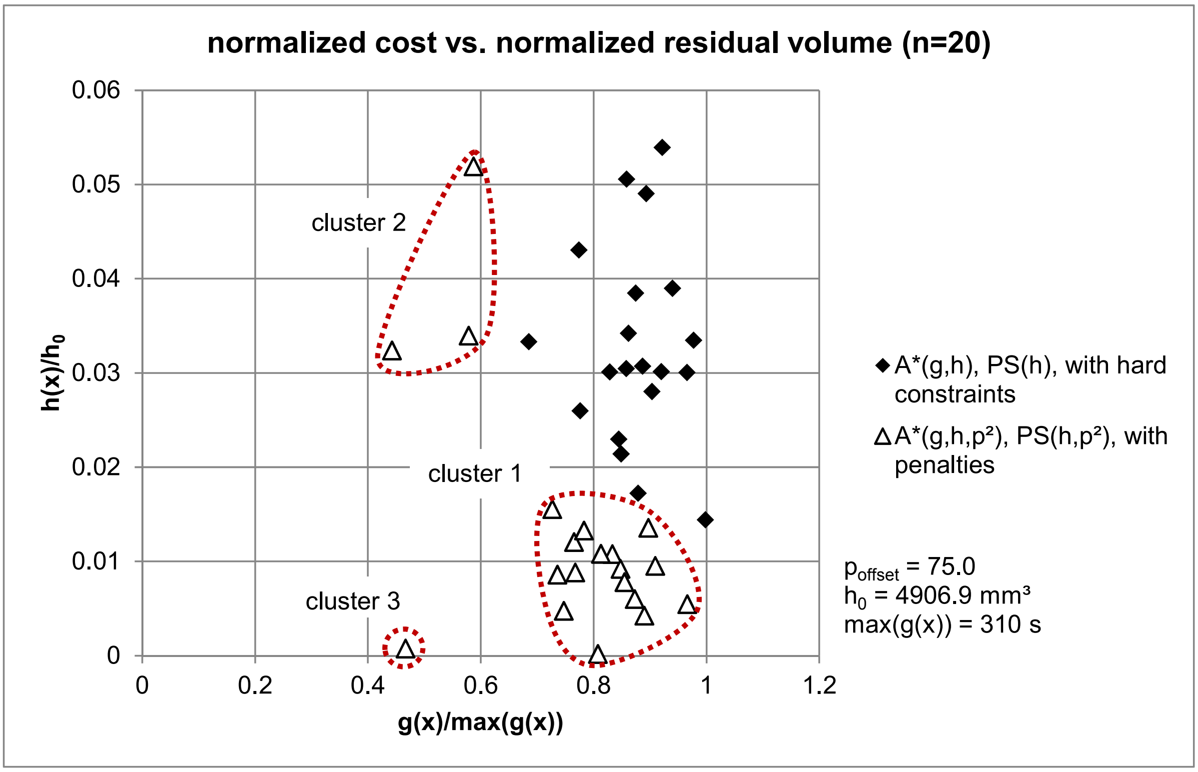


Figure 5-7: Approximated cost vs. normalized residual volume (error value) for A-PS with hard constraints (M1) and A*-PS with penalties (M2) for the part, shown in Figure 5-6.*

The detailed result statistics can be found in Table 5-2. The columns are the same as in Table 5-1. The rows represent the different methods and clusters.

Table 5-2: Result statistics by methods, best and worst solutions for h and f, average values and standard deviations

		best min(h)	best min(f)	worst max(h)	worst max(f)	average	standard deviation
method M1	h [mm³]	70.7890	84.5702	264.6040	264.6040	161.0504	50.4058
	h_{res} / h_0	0.0144	0.0172	0.0411	0.0411	0.0328	0.0103
	g [s]	309.4860	272.4690	285.7190	285.7190	271.2112	22.3608
	p [mm³]	-	-	-	-	-	-
	f	190.1380	178.5190	275.1620	275.1620	216.1308	26.7910
method M2 overall	h [mm³]	0.8686	3.6637	254.7660	254.7660	63.6389	60.1405
	h_{res} / h_0	0.0002	0.0007	0.0519	0.0519	0.0130	0.0123
	g [s]	250.4040	144.7020	182.0320	182.0320	237.0982	43.3952
	p [mm³]	0.0000	0.0000	3.4403	3.4403	0.1941	0.7509
	f	125.6360	74.1829	305.2350	305.2350	158.4700	42.8260
method M2 cluster 1	h [mm³]	0.8686	0.8686	76.2303	66.4211	43.0520	18.9581
	h_{res} / h_0	0.0002	0.0002	0.0155	0.0135	0.0088	0.0039
	g [s]	250.4040	250.4040	225.3160	278.0630	256.1554	20.9246
	p [mm³]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	f	125.6360	125.6360	150.7730	172.2420	149.6036	12.1502
method M2 cluster 2	h [mm³]	158.8820	166.6340	254.7660	254.7660	193.4273	43.4883
	h_{res} / h_0	0.0324	0.0340	0.0519	0.0519	0.0394	0.0089
	g [s]	137.3760	179.3680	182.0320	182.0320	166.2587	20.4521
	p [mm³]	0.4414	0.0000	3.4403	3.4403	1.2939	1.5284
	f	223.3240	173.0010	305.2350	305.2350	233.8533	54.4953
method M2 cluster 3	h [mm³]	3.6637	-	-	-	-	-
	h_{res} / h_0	0.0007					
	g [s]	144.7020					
	p [mm³]	0.0000					
	f	74.1829					

Comparing method M1 and method M2 overall, method M2 shows better best and worst solutions than method M1. Also, the average values are better too. However, due to the existence of the three clusters, method M2 shows overall higher standard deviation values. Considering cluster 1 of method M2 compared to method M1, it can be seen that method M2 produces better results for both *min(h)* and *min(f)*. Also the worst solutions (with *max(h)* and *max(f)*) have lower values than those of method M1. The average values and standard deviation values are generally lower than those of method M1. It is remarkable that the penalty is zero for all solutions within cluster 1. Considering only the cluster 2, it can be seen, that the best solutions of method M2 are worse than those of method M1. However, the worst solution of method M2 in cluster 1 is still better than the worst of method M1. Some of the solutions within cluster 2 show a non-zero value penalty value. Cluster 3, consisting of only one solution is much closer to the goal state than any other result.

Overall it is reasonable to conclude that method M2 is able to solve more complex parts better than method M1 while still maintaining similar results as method M1 for simpler examples. The exceptional result in cluster 1 of method M2 indicates the potential for improvements using more advanced heuristics to drive the search process more closely and reliably to the desired goal state. Therefore, only method M2 is pursued further in this thesis.

5.2 Autonomous generation of machining plans

The previously presented results only considered the machining planning process itself. In this section, the process is extended to cover more aspects of design-to-fabrication. The typical application of this scenario is the autonomous fabrication of customized parts where a 3D model of a customized part is given as input and the output is the machining plan that is directly executed on the hardware.

To accomplish this scenario, the following operations have to be carried out

1. Specifying the customized part geometry,
2. selection of a fitting workpiece,
3. generation of machining plan using method M2 and
4. execution of the machining plan on the hardware.

First, the customized part geometry needs to be specified. This is done by loading a STEP file into the GUI of the Spatial Grammar Machining Planning (SGMP) system. This GUI, based on the OpenCascade sample application Viewer3d (OPEN CASCADE S.A.S. 2011), is shown in Figure 5-8.

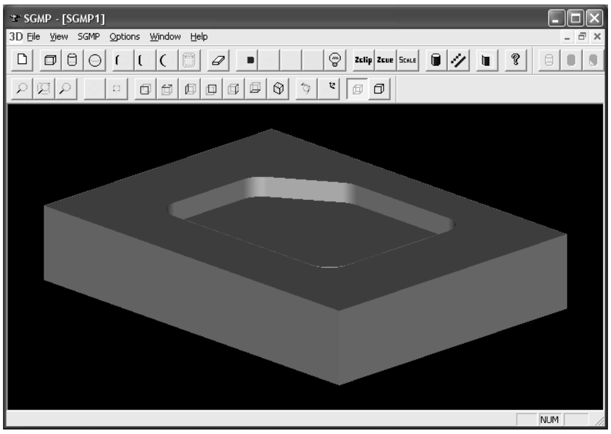


Figure 5-8: Part to be fabricated within the software for machining planning

In the central area of the window, the part geometry can be inspected using the tools and menus provided by the Viewer3d application.

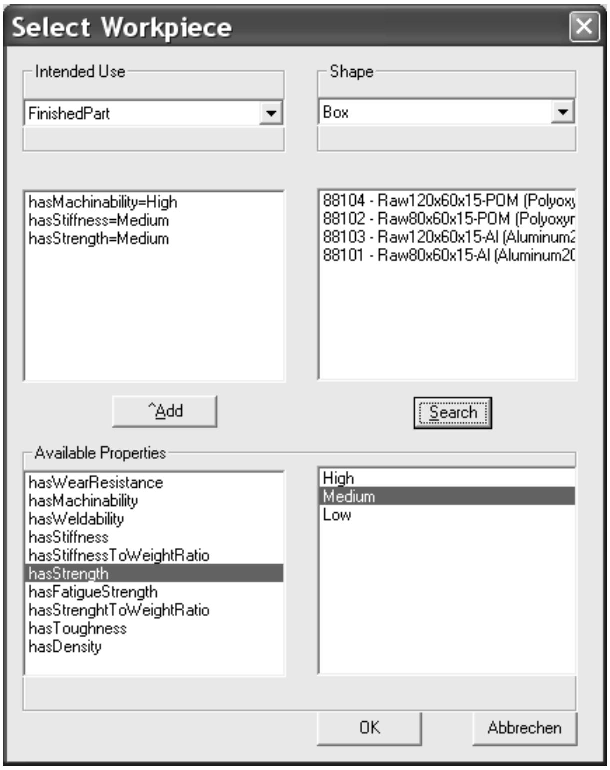


Figure 5-9: Workpiece selection dialogue, querying the workpiece selection ontology (SHEA et al. 2010).

In the second step, a workpiece for the fabrication of the part must be selected. This choice does not only include geometric considerations but also requirements regarding the quality and properties as well as machining process constraints such as machinability of the workpiece as specified by the user. For the selection of the workpiece, an ontology is used that is able to propose one or several alternative workpieces to the user (SHEA et al. 2010). This specification and results returned by the ontology are shown in Figure 5-9.

At the top of the dialogue, the user can specify the use of the part. Choices are FinishedPart (highest accuracies), FormPrototype (good shape creation, material may differ), FunctionalPrototype (good mechanical strength). Further, the shape of the workpiece can be specified, either box shaped or cylindrical.

At the bottom of the dialogue, available properties can be selected and their value specified and added to the list of requirements. After the search, the ontology proposes several workpieces on the right list. In the example shown, a workpiece for a finished part with a box shape, high machinability, medium stiffness and medium mechanical strength is required. The ontology proposes four different workpieces as shown on the right list in Figure 5-9. After selecting one workpiece from the list, the shape of the workpiece is created within the GUI of the SGMP system as shown in Figure 5-10.

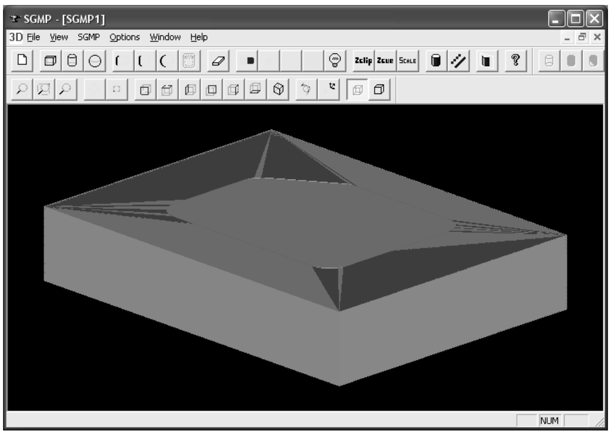


Figure 5-10: Desired part and shape of the selected workpiece in the workspace.

Here, the workpiece is displayed in gray while the customized part in dark gray can partially be seen. From the part shape and workpiece shape, the material that needs to be removed is calculated (TRV) as shown in Figure 5-11. The TRV is displayed in dark gray and the non-open faces (NOF) are displayed in light gray.

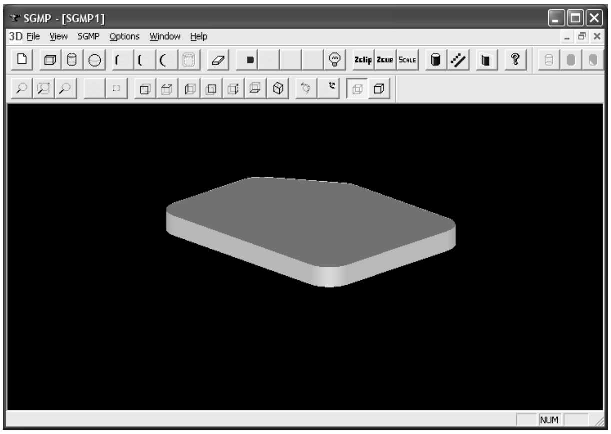
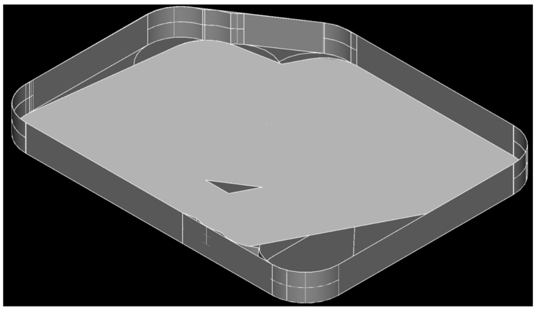
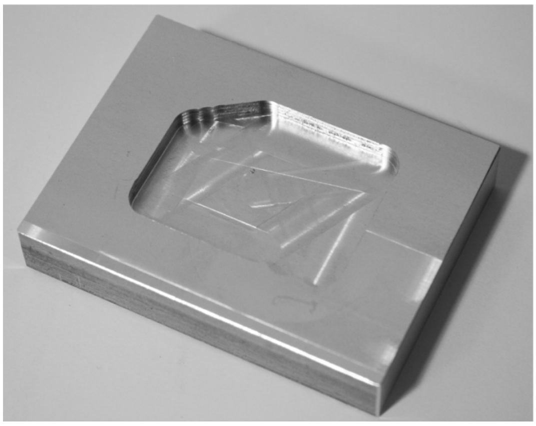


Figure 5-11: Total Removal Volume (TRV) calculated from the part and workpiece shape.

With this data, the machining planning process using SGMP can be started. After the planning, the CNC code can be generated. Since every rule and RV corresponds to a specific machine tool, tool and machining instructions, the CNC code can directly be derived from the sequence of rule and RV applications. Once the CNC code has been generated, it can be transferred over a network connection to the CNC machine tool where it is executed. Figure 5-12 displays the TRV after decomposition, i.e. after the planning process and a part that was fabricated using the CNC code generated by the SGMP system.



a



b

Figure 5-12: Geometric representation of a generated process plan (a) (M2, h = 24.706, g = 288.878, p=0 f=156.792, computation time t=2147s, colors modified for clarity) and fabricated part (b)

5.3 Machining planning for multi-body TRVs

For most engineering parts, more than a single protrusion has to be machined. Instead a customized part may consist of several areas that create not a connected Total Removal Volume (TRV) but several disjoint bodies. Figure 5-13 shows such a part.

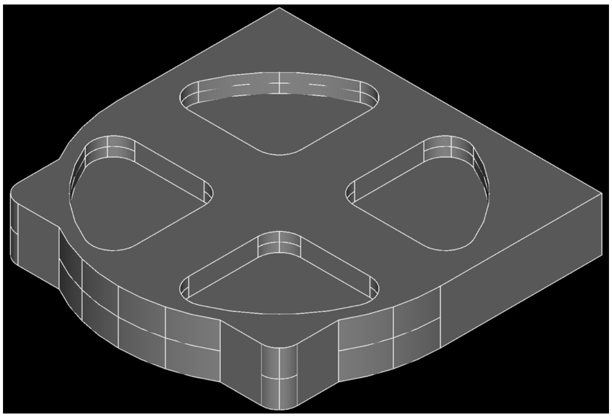


Figure 5-13: Part that creates several TRVs.[1]

Here several protrusions must be machined from the stock. However, calculating the TRV from the stock and part shown in Figure 5-14, the TRV yields to the shape with multiple bodies shown in Figure 5-15.

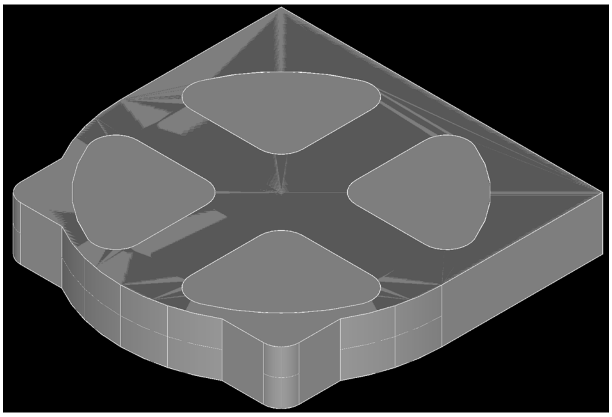


Figure 5-14: Part (dark gray) and workpiece (gray).

[1] Example part simplified from motor casing, kindly provided by Brennand Pierce, Institute for Cognitive Systems, Technische Universität München

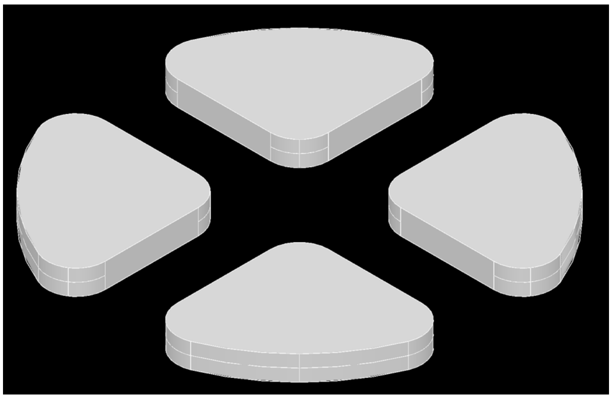


Figure 5-15: Resulting, multiple bodies within one TRV.

The decomposition of all bodies of the TRV within a single search process is very complex and prone to fail by not reaching all bodies during the search. This is due to the requirement, that the toolpath must be continuous. Therefore, the search would have to relocate the tool to each body of the TRV by itself and then completely remove the volume. To overcome this, the core functionality of the CAD kernel is used to identify the individual bodies shown in Figure 5-15. Then, the search can find a feasible machining process plan for each individual body. Once a feasible plan for each body has been generated, the machining plan for the complete part can be created by concatenating the machining plans for the individual bodies.

The process of this iterative decomposition is shown in Figure 5-16. From the part shape and the workpiece shape, the Total Removal Volume (delta volume) can be calculated. It consists of four separated bodies. A machining plan for each body is generated. The overall machining plan can be generated as a concatenation of the individual plans in the sequence of the search. Currently the sequence of plan generation for separate bodies is arbitrary but can be extended to a ranked order by volume. Since every part of the TRV can be decomposed into a feasible machining plan with safe and feasible approach and retract positions, the whole TRV can be machined by the sequence of plans generated for the individual parts of the TRV. The overall machining plan made up of the individual machining plans is graphically shown in Figure 5-17.

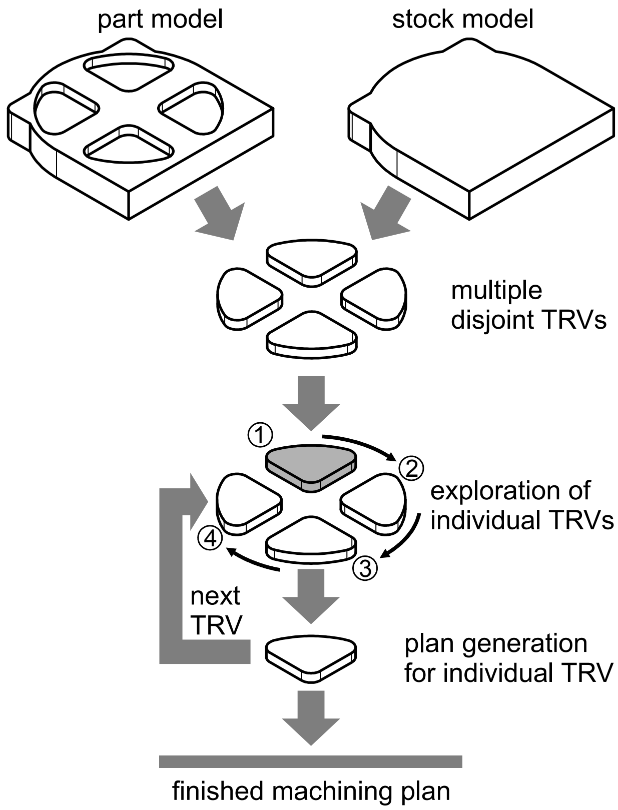


Figure 5-16: Machining planning process for multiple, disjoint TRVs.

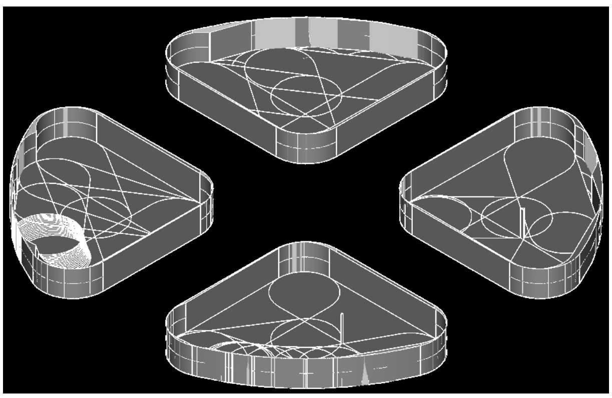


Figure 5-17: Graphical representation of the machining plan for multiple TRVs (colors modified for clarity) (method M2)

5.4 Use of multiple tools

The previously presented scenario used only a single tool. However, in industrial practice, multiple tools are used, usually a bigger tool for roughing and a smaller tool for finish machining and details of the part.

To reflect the necessity for using multiple tools in the planning system, new Removal Volumes for different tools can be introduced. However, to take advantage of these, it must be ensured that:

- The cost caused by tool changes are taken into account during the planning process and
- the constraints are maintained, such that tools are retracted prior to changing the tool.

If the search attempts to apply a tool change alone, it would never be accepted since it only causes cost and therefore increases the objective function value. Therefore, this rule must be combined with other rules that have a positive effect, i.e. lowering the objective function. This combination must comprise rules that can directly take advantage of the tool-change and thus are able to reach overall a lower objective function value than other rules.

In the implementation of the planning system, this has been accomplished by the following measures:

(a) The rule “ChangeTool” has been implemented providing the base functionality and allowing for a safe reatraction of the tool before initiating the exchange of the tools.
(b) The constraint “DifferentTool” has been introduced, ensuring that no tool change between the same tools is attempted.
(c) The application of Removal Volumes is constrained to use the same tool as the previous one through the constraint “SameTool”
(d) The new rule class “Rule Sequence” has been implemented. This allows defining a sequence of rules to retract the tool, change the tool and re-approach the part safely. Of course, rule sequences can also be used for other purposes.
    (1) The RuleSequence “ChangeTooltoD10”, performing a tool change to a D10 endmill, applying a cylindrical Removal Volume and a slot shaped Removal Volume is implemented.
    (2) The RuleSequence “ChangeTooltoD15”, performing a tool change to a D15 endmill, applying a cylindrical Removal Volume and a slot shaped Removal Volume is implemented

By using sequences of rules, the rule “ChangeTool” can be applied though it only causes cost and would therefore never be applied alone during search by any opportunistic search method such as the A* search method.

Figure 5-18 shows a part that can be best machined using multiple tools, namely a D15mm endmill and a D10mm endmill. Machining the complete protrusion using only the D10 endmill would be ineffective and result in a long machining time.

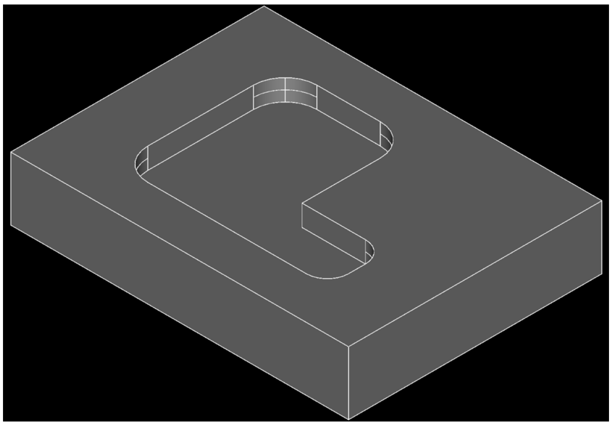


Figure 5-18: L-shaped part to enforce tool changes (colors modified for clarity)

Using a D15 endmill, only the shaded area shown in Figure 5-19 is accessible, therefore the part is not completely machinable using only this tool.

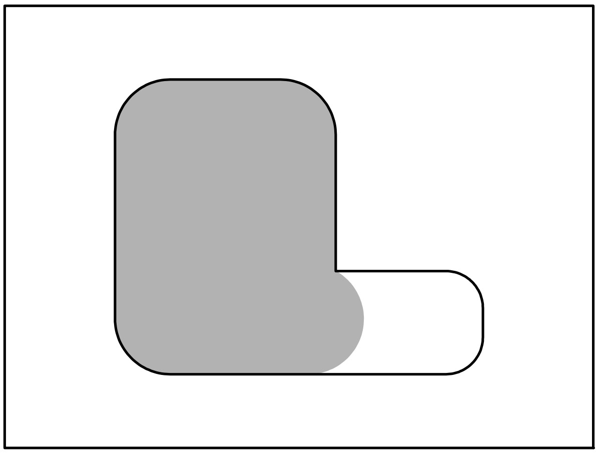


Figure 5-19: L-shaped part with accessible region for a D15 endmill (shaded area)

Figure 5-20 shows the TRV after a complete planning process. The SGMP system starts by applying the larger D15 endmill until all of the accessible area (cp. Figure 5-19) is removed. Afterwards, the system performs a tool-change and continues using the D10 endmill. Though the system is able to change both ways between the tools at any time, it only changes the tool from D15 to D10 once.

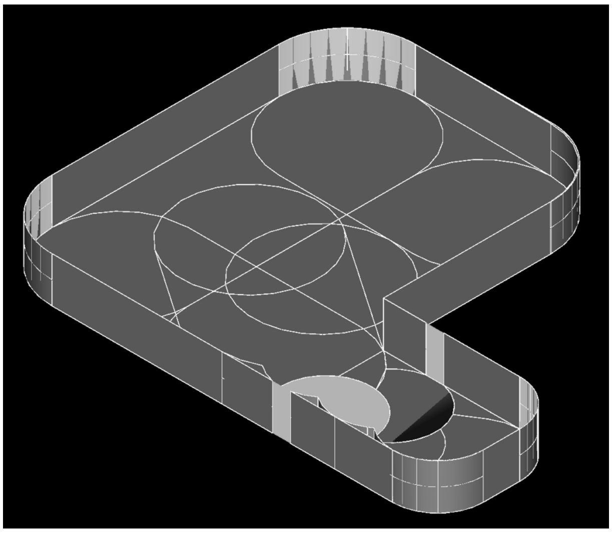


Figure 5-20: L-shaped part as planned for fabrication using two different tools (M2, h = 38.1213, g = 172.638, p=0.654078 f=116.633, computation time t=4150s) - (colors modified for clarity)

After the planning process, the CNC code was used to machine the part as shown in Figure 5-21.

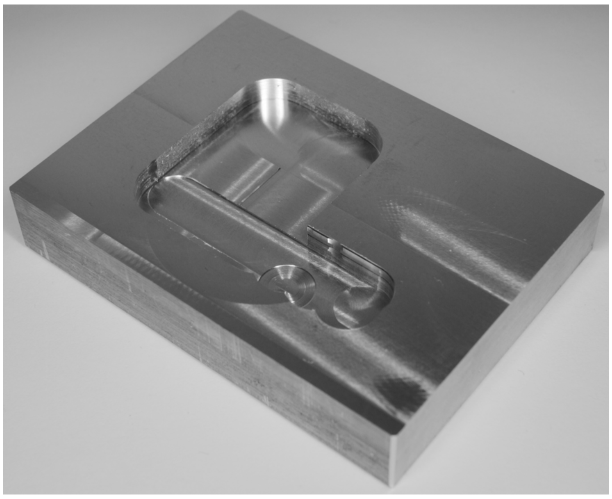


Figure 5-21: Machined part using the CNC code created by SGMP.

To compare the performance and to prove that the use of several tools is more effective, the planning process is carried out using only the D10 endmill. The resulting TRV is shown in Figure 5-22. Here, the values for the goal distance function *h* and the cost *g* are much higher than using multiple tools.

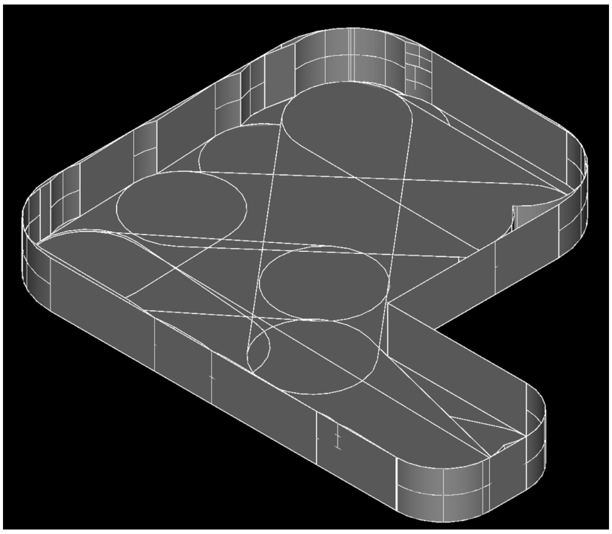


Figure 5-22: The same part using a single tool D10
(M2, h = 110.136, g = 355.674, p = 0, f = 232.905, computation time t = 3661s)
- (colors modified for clarity)

5.5 Reaction to (un-)availability of workpieces

During the operation of an autonomous design-to-fabrication system, the situation of missing workpieces can occur. This situation and actions of the system are presented in the following. After a plan for machining is generated and initiated, the system tries to find the workpiece with a box shape and dimensions of 80x60x15mm. For some reason, the workpiece is missing. However, alternative workpieces are available. Once again, the ontology for material selection can be used to find the next best workpiece for the fabrication of the customized part (Figure 5-23).

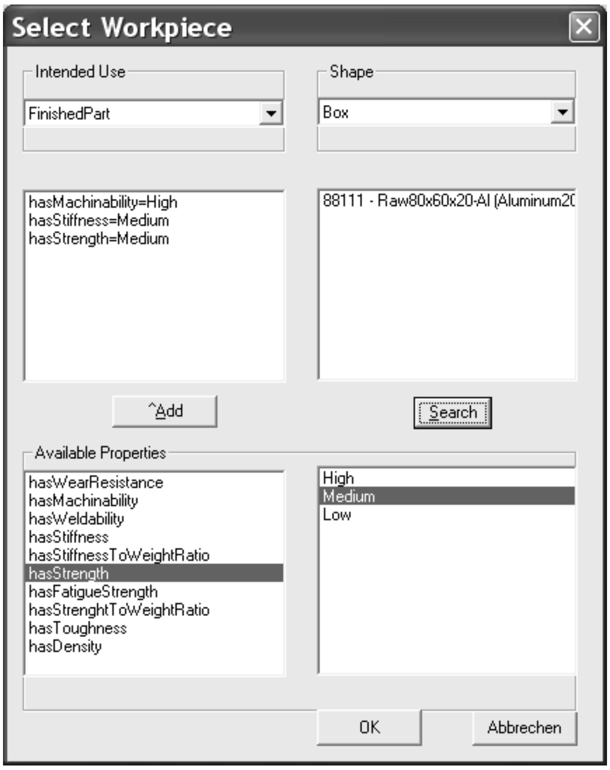


Figure 5-23: Workpiece selection where only one alternate workpiece is available

The selected alternate workpiece can then be used as input to the planning system. However, performing a complete replanning of the machining process is less effective. If a plan for the fabrication of the customized part with the original workpiece exists, the system needs only to generate a machining plan for fabricating the original workpiece from the alternate. The original workpiece is then treated as the part to be fabricated.

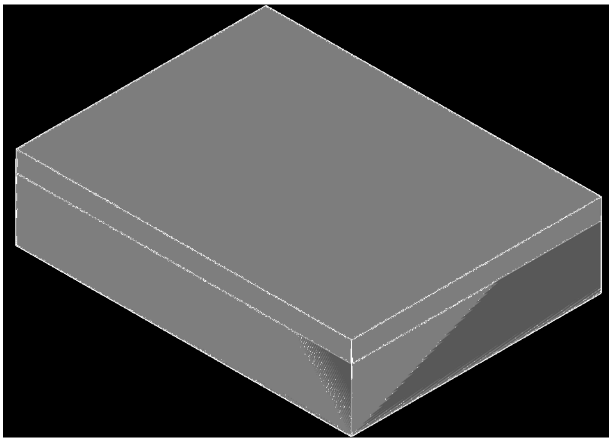


Figure 5-24: Total Removal Volumes of original (dark gray) and alternate (gray) workpiece (colors modified for clarity).

Figure 5-24 illustrates this situation where the original workpiece is loaded as the part (gray color) and the slightly larger alternate workpiece is loaded as the workpiece (gray). The resulting TRV is depicted in Figure 5-25.

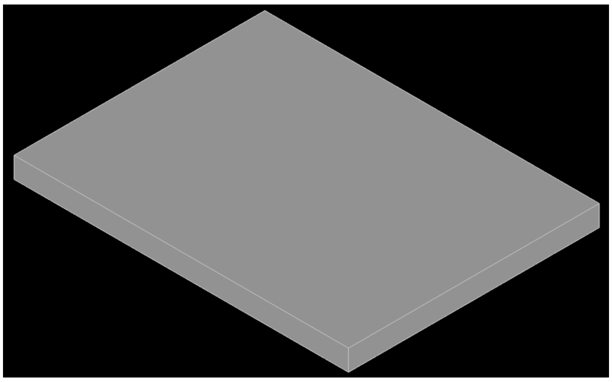


Figure 5-25: Resulting TRV (colors changed for clarity).

With this input, the SGMP system can plan the machining. The result of one planning run is shown in Figure 5-26.

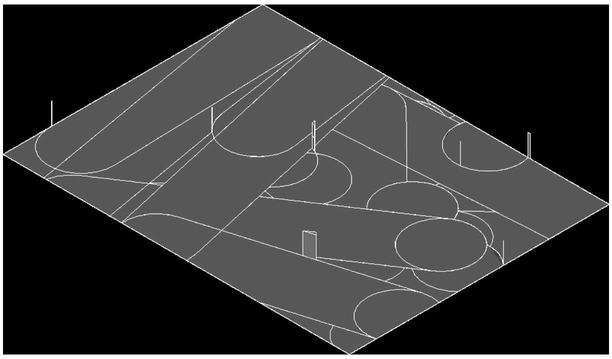


Figure 5-26: Decomposed TRV of the difference between the two workpieces (M2, h = 6.085, g = 609.405, p = 0, f = 307.745, computation time t=3225s) - (colors changed for clarity).

The original workpiece has successfully been machined from the alternate workpiece as shown in Figure 5-27.

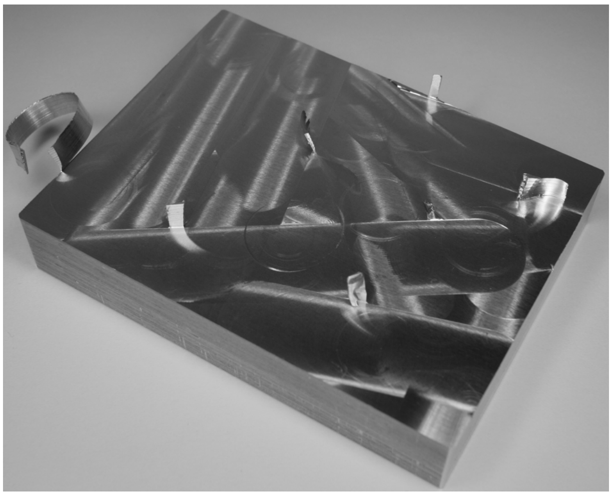


Figure 5-27: Machined original workpiece.

Using this workpiece, the original machining plan for the customized part can now be executed.

5.6 Reaction to tool failures

An important issue is the reliable operation of fabrication systems for autonomous design-to-fabrication. Ideally, the fabrication system is able to recover from failures. One typical failure during machining is tool failure. Here, the tool, used for cutting, is destroyed or made unusable which results in unwanted deviations of the workpiece shape. Figure 5-28 shows an example of a destroyed workpiece due to tool failure. Through increased temperature at the tool workpiece contact, the material is no longer completely cut but partially altered by plastic deformation. This results in deviations from the desired shape.

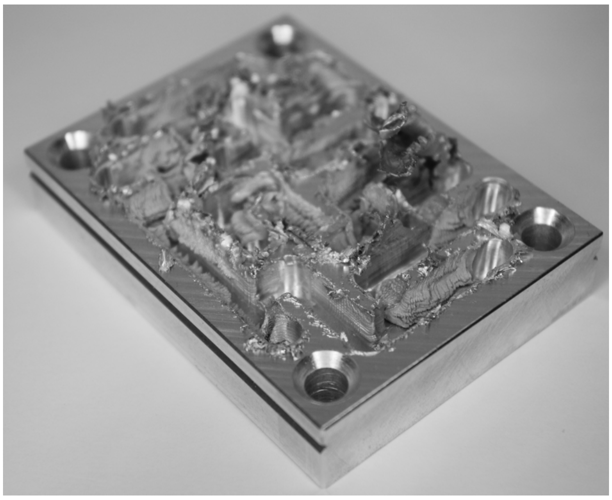


Figure 5-28: Destroyed workpiece due to tool failure

Figure 5-29 displays the timeline of such a situation and which measures should be taken by an autonomous fabrication system.

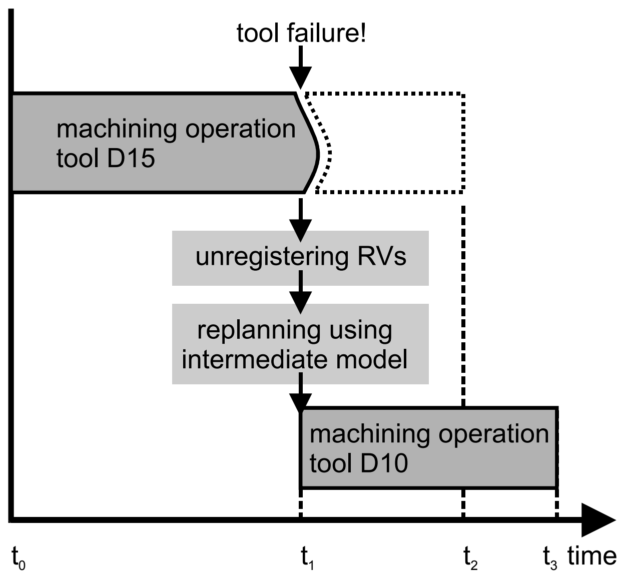


Figure 5-29: Timeline for machining operation with tool failure and reaction of the system

At time t_0, a machining operation begins using the tool D15. This operation is expected to last until t_2. However, at time t_1 (before t_2 is reached) a tool failure is detected and the machining operation is interrupted. Since the currently used tool must be considered as defective, it must not be used in following machining operations. Therefore, the knowledge model is updated by unregistering the Removal Volumes (RVs) and rule associated with this tool in the machining planning system. Since the machining planning system capitalizes on atomic machine tool motions, the Total Removal Volume is a representation of the actual workpiece shape during machining. Therefore, the machining operation can be replanned using the TRV at time t_1 for replanning the machining operation using a different tool, here the tool D10.

Figure 5-30 shows a graphical representation of the machining plans generated with a simulated tool failure (a) and without (b). The different size of the shaded area on the bottom is due to the numerical resolution of the visualization. Otherwise, the resulting part shape is almost identical. Though the machining takes more time, until t_3, the design-to-fabrication system is still able to produce the requested part.

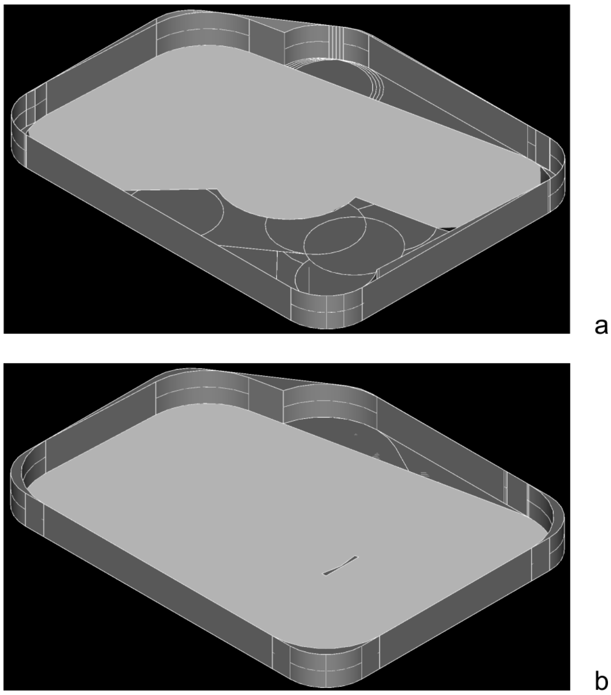


Figure 5-30: Comparison of results
with simulated tool failure (a) (M2, h=133.889, g=195.05, p=0, f=164.469, computation time t=522s)
and without (b) (M2, h=151.089, g=155.351, p=0, f=153.22, computation time t=437s)
- (colors enhanced for clarity)

5.7 Discussion

In this section several scenarios and results from the application of the SGMP method have been presented: machining planning for general parts, autonomous machining planning involving material selection and CNC fabrication, machining planning for parts with a disjoint TRVs, use of multiple tools, reaction to (un-)availability of workpieces and reaction to tool failures. The scenarios presented demonstrate the flexibility and the potential of the approach.

Reacting to tool failure demonstrates the ability of the machining planning method to handle dynamic changes in the fabrication system, reflected by adding and removing Removal Volumes and rules to and from the grammatical framework. The planning system can then instantly take advantage of new rules and Removal Volumes. This can also be used to add new resources to a fabrication system namely machine tools or new and changed tools. Further use of this can be to consider scheduling aspects by reflecting the availability of machine tools online within the grammar.

The implementation is capable of creating valid machining plans for customized parts based on their CAD geometric representation. It capitalizes on a strict object-oriented framework and powerful patterns that allow for straight forward extension of the grammar by creating new rules, Removal Volumes and constraints. The planning can be carried out for single or multiple TRVs (machining volumes). The method in implementation can capitalize on the availability of multiple tools and thus can use the tools that lead to best machining performance and least time for machining. The current implementation is limited to 2.5D parts that can be milled using end-mills with a pre-specified approach direction. However, the method is extendable to more complicated machining processes such as turning and multi-axis machining.

The results demonstrate the general feasibility of the approach but also illustrate potential for improvement. First, the search method, in terms of both result quality and computation time, must be further enhanced by adding heuristics, based on best practice knowledge, that help guide the search process better towards globally optimal solutions. It might also be beneficial to use more powerful heuristic search methods as well as applying an iterative optimization method for re-optimizing previously applied rules.

5.8 Conclusion

To create an effective and automated system for machining planning, the steps of shape recognition, mapping shapes to machining operations and scheduling operations need to be integrated. The method and implementation presented achieves this by using a combination of a spatial grammar and heuristic search methods that allow for spatial reasoning in terms of which operations are required to remove the correct volume from the stock while being able to directly generate the specific CNC instructions to execute the plan directly on the machine tool. The bottom-up and process-based approach, using the CNC instructions and their semantics to build up a library of manufacturing knowledge and capabilities makes the system highly adaptable and extendable. Since the rules and vocabulary build on machining operations and their instructions, they are associated directly with the shape. In contrast to feature-based approaches that use a taxonomy of symbols, the approach presented does not require grounding by mapping these symbols to actual machine operations. Further, the fundamental constraints on rule application are formalized with respect to the allowed spatial relations between the Removal Volume (RV) and the Total Removal Volume (TRV) and the allowed transformations, based on machine and process capabilities, during rule application. By changing the rule set and vocabulary of the grammar during runtime, changes in the live manufacturing system can be reflected. Overall, the scenarios presented demonstrate the feasibility of the approach.

The next section investigates more advanced search methods to improve the results.

6 Advanced heuristic search methods for Spatial Grammar Machining Planning

Planning the machining of a specific region of a workpiece has been identified before as a covering salesperson problem (ARKIN & HASSIN 1994). Similar to the travelling salesperson problem, an agent should travel through a region, covering it entirely while minimizing the travelled distance. As proved by ARKIN et al. (2000) this problem is NP-hard, i.e. it cannot be solved in polynomial time. However, numerous algorithms have been developed to approach the problem of automatic tool-path generation for CNC machining of pockets. Most approaches create the toolpaths either by "zig-zagging" over the region or by using offsets of the outer contour of the pocket (HELD 1991). Both strategies solve the problem for a single pocket. However, these algorithms only solve this problem, thus their application requires the identification of situations where they can be applied as for example feature recognition or feature mapping. Also, the algorithms do not incorporate inherent process constraints such as requiring continuous toolpaths and do not have simulation capabilities to validate the generated plans.

In contrast to the algorithms briefly described, the SGMP system is not designed to solve the specific case of machining a single pocket. Since it is built on the fundamental machine tool and tool capabilities, the method by design is not limited to 2.5D milling of pockets but can also be applied to multi-axis machining and different machining processes in theory, to every machinable part and workpiece. However, quality and effectiveness of the generated plans is not guaranteed nor might be achievable due to the high complexity of the problem, especially when using multiple tools.

The subject of this section is the improvement of the search method for better plan quality. After a presentation of the advanced search methods, results using A* with Pattern Search and different penalties will be presented. Simulated Annealing (SA) is applied to state space search, i.e. selecting the rules to apply. Last, the concept of re-optimization of machining plans is presented.

6.1 Simulated Annealing

Since the development of Simulated Annealing (SA) (LEVY et al. 2003) from the original Monte-Carlo algorithm, it has gained a lot of attention in research, due to the ease of implementation and its robust and statistical behavior, resulting in the ability to recover from local optima. SA has been successfully applied to design synthesis (CAGAN & MITCHELL 1993) as well as process planning and scheduling (LI & MCMAHON 2007). The name Simulated Annealing stems from the physical analogy of annealing metal, i.e. treating the metallurgic properties by heating and cooling the material according to a specific schedule. The material then drops into specific crystal configurations that can be identified as local or global minima of energy of the crystals. SA attempts to achieve the same with the objective function. Consequently SA also features a control parameter called temperature *t*. The formulation of Simulated Annealing in this work is based on the work by SHEA (1997) and has been adapted to this new search problem. The SA algorithm implemented in this work is given in Table 6-1.

Table 6-1: Basic Simulated Annealing algorithm in pseudo code.

```
Initialize temperature t with some t_0
while halt criterion is not met
    while termination criterion not met
        generate new solution candidate
        evaluate candidate quality f_{j+1}
        calculate change in objective function Δf = f_j - f_{j+1}
        if change of objective function Δf < 0
            accept solution candidate
        else accept with probability q(Δf, t)
        else reject solution
        if candidate is better than best solution
            save candidate as best solution
        update temperature
        check termination criterion
        check halt criterion
```

The probability q that determines the acceptance of a solution candidates in case $\Delta f \geq 0$ is defined as

$$q(\Delta f, t) = e^{-\frac{\Delta f}{t}}. \qquad (20)$$

The temperature t has to be lowered slowly enough and the initial temperature t_0 has to be chosen appropriately, trading of search time by using a high initial temperature and risking to not finding the global optima by choosing a too low initial temperature. Choosing the initial temperature and the cooling schedule is a major challenge in applying SA. To overcome this, Lam and Delosme (LAM & DELOSME 1988) developed an efficient cooling schedule based on statistics gathered during the search process.

The theory behind the Lam-Delosme schedule states that a solution can be found if the thermodynamic equilibrium is met. To achieve this, a specific acceptance ratio α_{actual} is maintained by adjusting the temperature accordingly. The acceptance ratio is defined as

$$\alpha_{actual} = \frac{\#accepted\ moves}{\#all\ moves}. \qquad (21)$$

The schedule presented here has been modified by Swartz and Sechen (HUMMEL 1989) from the original Lam-Delosme schedule. Depending on the current number of iterations, the target acceptance ratio α is calculated and the temperature adapted to meet this acceptance ratio. An example for the course of the target acceptance rate is displayed in Figure 6-1.

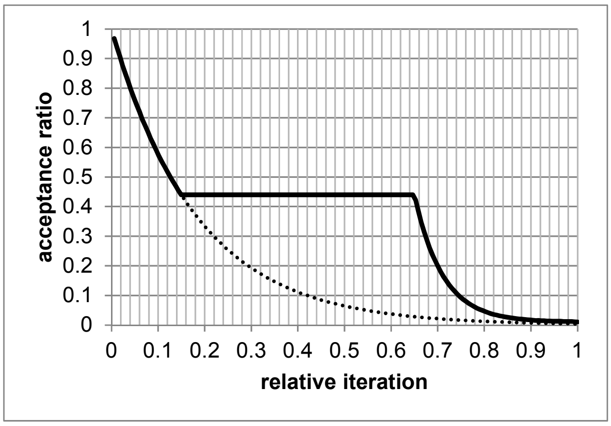


Figure 6-1: The course of the target acceptance ratio α using the modified Lam-Delosme schedule (HUMMEL 1989).

The acceptance ratio begins with a high value of about 0.96, therefore almost all inferior solutions are accepted at the beginning of the search. The target acceptance rate is lowered quickly during the first 15% of the iterations. The curve approaches zero asymptotical towards the last iterations (dashed line). Then, the target acceptance ratio is kept at a stationary value of α=0.44. After 65% of the iterations have been completed, the acceptance rate is lowered again until a minimum value of α=0.01 is reached.

The temperature for each new move $i+1$ in each iteration can be calculated using

$$t_{i+1} = \left(1 - \frac{\alpha_{actual} - \alpha}{K}\right) t_i \qquad (22)$$

where K is a damping factor. The actual acceptance ratio, α_{actual}, is calculated during the search process. With this schedule, the temperature is adjusted to the specific problem and to the progress of the search. The statistic nature of SA also results in a high computing effort due to the high number of iterations and moves.

Generating the candidate solutions, also called neighbors, is a crucial step during search. The choice is problem specific and depends on the type of search space, whether it is discontinuous (rule-based) or continuous (parameter-based).

Continuous search space

Within a continuous search space, several variables can be modified continuously to find a solution. Generating the neighbor can either be obtained by some kind of heuristic that gives directions a priori on the search, or purely at random by modifying one, several or all parameters at the same time by adding or subtracting some random value. However, the kind of randomness is important in this case for changing the variables randomly. It is suggested to use a Gaussian or Cauchy random distribution (MICHALEWICZ & FOGEL 2004). It is not advised to use a uniform random distribution. Otherwise the search can alternate between a small number of very similar solution candidates since it is likely, due to the uniform distribution, that a step in a specific direction is revoked by another step (with similar length) in the opposite direction.

Discontinuous search space

For a discontinuous search space, encountered in grammar based searches, the choice of candidates can rely on a probability distribution over all rules. A candidate is generated by applying a rule to the current state or configuration. The choice of the rule itself depends on a static probability of picking the rule. However, such a static rule probability can result in slow search progress since rules that are helpful at the beginning of the search might hinder the search at later iterations and vice versa. Further, the rule probabilities are tuned to fit one specific problem. HUSTIN proposed to not use static rule probabilities but to recalculate the probabilities during the search to overcome this problem (HUSTIN 1988). A quality factor is introduced and defined as

$$Q_M = \frac{\sum_{accepted\ moves} size(M)}{\#\ attempted\ moves\ M}\,. \qquad (23)$$

The quality of each move (which corresponds to one rule application) equals the sum of all accepted changes to the objective function (here size(M)) divided by the number of the attempted moves M.

The probability of selecting a move or rule equals then the quality of the rule divided by the sum of all qualities, thus

$$P_M = \frac{Q_M}{\sum_i Q_i}\,. \qquad (24)$$

For the outer loop (discontinues) search, Simulated Annealing (SA) has been implemented using Hustin moves (HUSTIN 1988) and a tabu list for already tested rules. Through the use of a tabu list, the application of a rule to a state is attempted only once, preventing ineffective re-applications of the same rule to the same state.

The search procedure is shown in Table 6-2.

Table 6-2: Simulated Annealing outer loop (discontinues) search procedure

Start iterations
  Reset rule policy
  Start moves
    Select rule
    If rule has been tried at this state, skip it
    Apply rule
    If df < 0
      Accept
    Else accept with probability
    Update policy
  End moves
End iterations

6.2 Method studies

In the following, studies of the search methods and results are presented. Figure 6-2 shows the structure of the search problems within SGMP.

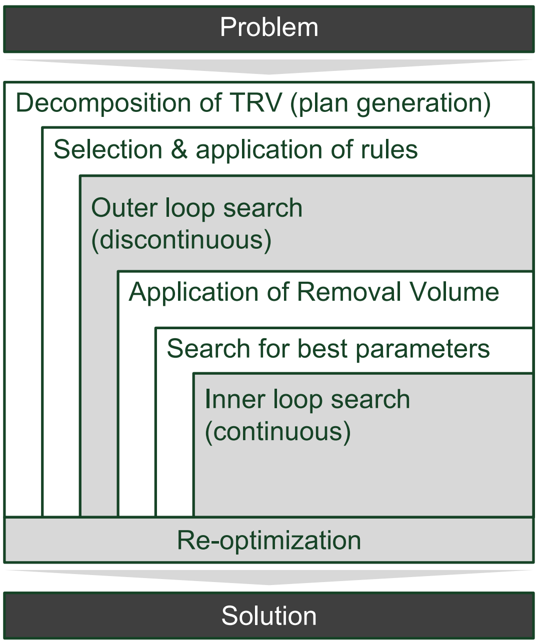


Figure 6-2: Structure of search problems within SGMP

At the top, the problem is formulated and passed to SGMP. The primary task is the decomposition of the TRV to generate a feasible machining plan. Within this decomposition, the rules need to be selected and applied to change the topology of the candidate solution. To do this, SGMP uses an outer loop search for this discontinuous problem. If a rule features a Removal Volume, its parameters need to be optimized. To do so, the continuous inner loop search is used to find good solutions for newly added elements of the topology. Optionally, the generated plan can be re-optimized as further described in Section 6.2.4. After finishing the decomposition or the re-optimization, the solution, i.e. the machining plan that has been generated, can be executed on the hardware.

Table 6-3 shows the different combinations of search methods that are applied within SGMP. Method M2 is using A* for the outer loop and Pattern Search with penalties in the inner loop as presented in Section 4. The results obtained using this combination are used as reference for the other search method combinations and are presented in Sections 6.2.1 & 6.2.2. Method M3 is Simulated Annealing with Hustin moves for the outer loop and Pattern Search with penalties on the inner loop. The results obtained using this combination of methods are presented in Section 6.2.3. Method M4 uses A* on the outer loop, Pattern Search with penalties on the inner loop and Simulated Annealing for re-optimization of the solution. The findings of this method are presented in Section 6.2.4.

Table 6-3: Portfolio of search methods applied.

Method	**Outer loop**	**Inner loop**	**Re-optimization**
Method M2 (4.5.3)	A*	Pattern Search w/ penalties	-
Method M3 (6.2.3)	Simulated Annealing w/ Hustin moves	Pattern Search w/ penalties	-
Method M4 (6.2.4)	A*	Pattern Search w/ penalties	Simulated Annealing

To allow for a comparison with the results presented in Section 5, the same example part shown in Figure 6-3 is used.

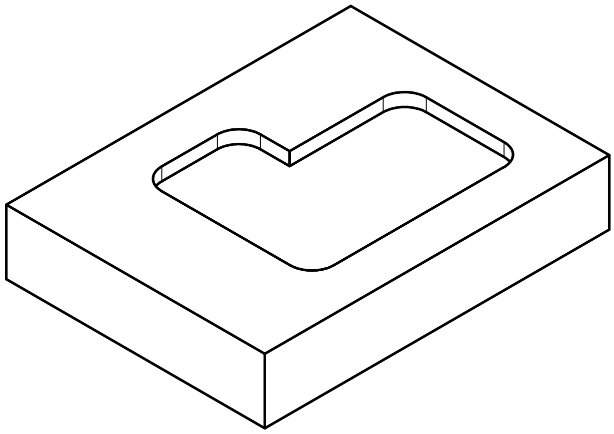


Figure 6-3: Example part with an L-shaped depression.

In the following the computation results and specific implementations for each search method are presented.

6.2.1 Behavior of A* for outer loop search

In this section the behavior of the A* search method (method M2) is investigated more closely. One complete search for the part shown in Figure 6-3 is displayed in Figure 6-4. The figure shows the values of the heuristic functions *h(x)* residual volume, *g(x)* cost and *f(x)* objective function over the A* iterations. The number of evaluated states is limited to 1000 states such that 334 iterations were run (branching factor 3). In the first iterations, *h(x)* drops steeply below 500 while cost is almost linearly accumulated during the first ten iterations. Then the search is almost stationary until a drop in *h(x)* is reached at iteration 20. Another drop can be seen at iteration 35. After this, the search remains mainly stationary. The fluctuation in *h(x)* between 110 and 190 iterations indicates that the search still attempts to find better solutions. After iteration 190, the change in the objective function becomes very small.

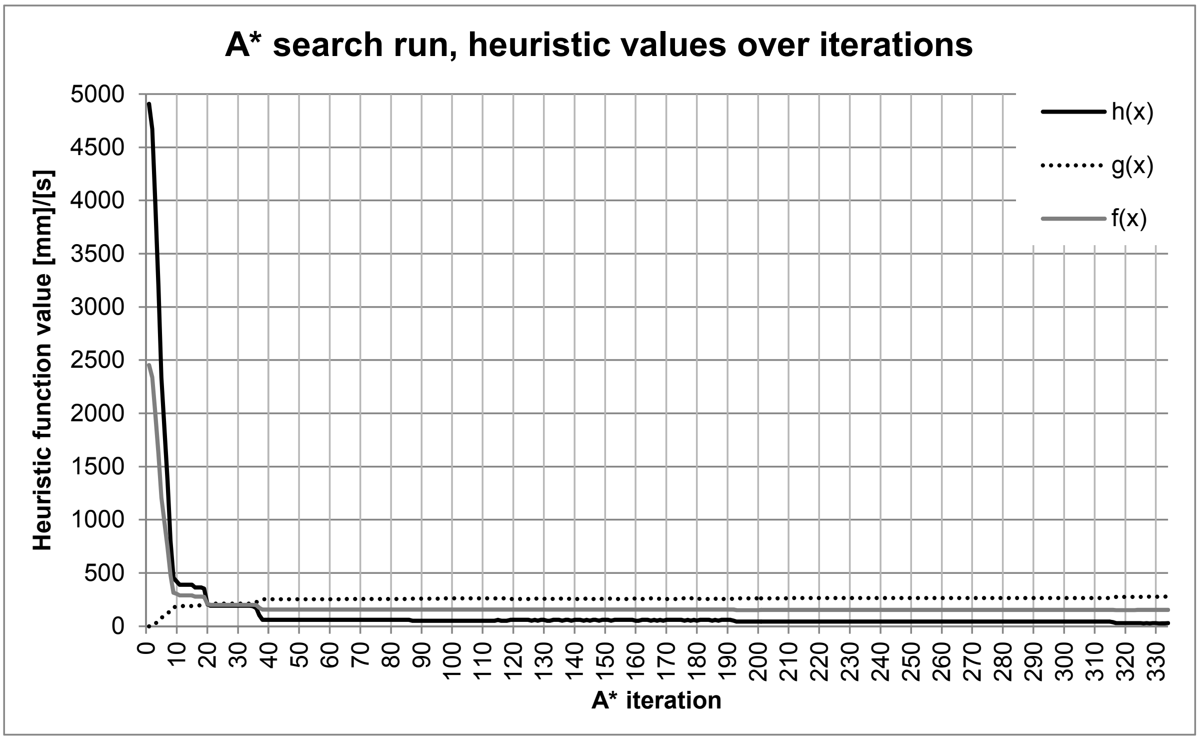


Figure 6-4: A search run on the part shown in Figure 6-3, heuristic values over iterations.*

6.2.2 Behavior of A* for outer loop search with variation of penalty offsets

To analyze quantitatively A* with Pattern Search (M2), several runs with different penalty offsets using the example part described are performed. In A*, a maximum number of 1000 states is evaluated. The comparison of A* with Pattern Search with penalty offsets of $p_{offset}=75$ and $p_{offset}=10$ is shown in Figure 6-5. Through a higher p_{offset} value, the search less likely accepts penalties at all (cp. eq. (13) & (19)). The diagram shows the normalized residual volume and the approximated cost for machining the part with the ideal solution in the lower left corner. The search with $p_{offset}=75$ shows in general good behavior and most solutions are near a normalized residual volume of 0.01 and normalized cost of 0.8. A few solutions reach lower cost. However, most of these only solve the problem partially, i.e. the residual volume is up to five times larger. Only one exception with almost no residual volume, i.e. $h=0.8686$ was generated. The solutions obtained with $p_{offset}=10$ form a compact cluster in a similar area to those with the larger penalty offset.

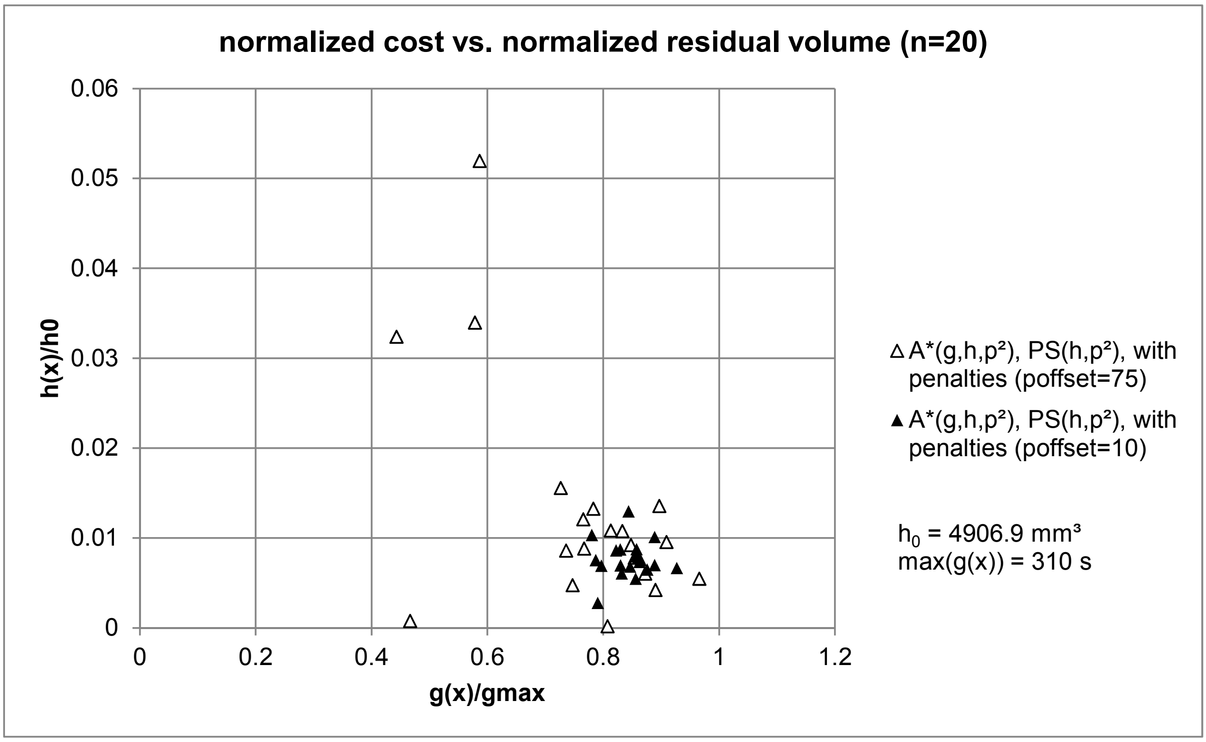


Figure 6-5: Approximated cost vs. normalized residual volume (error value) for A-Pattern Search with penalties (M2) and different penalty offsets.*

The detailed result statistics are shown in Table 6-4. The columns show solutions for minimum residual volume *min(h)*, minimum goal function value *min(f)* and maximum residual volume *max(h)*, maximum goal function value *max(f)* as well as average values and standard deviations of the solutions.

Table 6-4: Result statistics by methods, best and worst solutions for h and f, average values and standard deviations. Values for p_{offset}=75 have been presented in Table 5-2.

		best min(h)	best min(f)	worst max(h)	worst max(f)	average	standard deviation
A* -PS* (M2) $p_{offset}=75$	h [mm³]	0.8686	3.6637	254.7660	254.7660	63.6389	60.1405
	h_{res} / h_0	0.0002	0.0007	0.0519	0.0519	0.0130	0.0123
	g [s]	250.4040	144.7020	182.0320	182.0320	237.0982	43.3952
	p [mm³]	0.0000	0.0000	3.4403	3.4403	0.1941	0.7509
	f	125.6360	74.1829	305.2350	305.2350	158.4700	42.8260
A* -PS* (M2) $p_{offset}=10$	h [mm³]	13.5313	13.5313	63.5919	63.5919	37.4282	9.8656
	h_{res} / h_0	0.0028	0.0028	0.0130	0.0130	0.0076	0.0020
	g [s]	245.1050	245.1050	261.5800	261.5800	261.9818	11.3671
	p [mm³]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	f	129.3180	129.3180	162.5860	162.5860	149.7049	7.6878

Comparing the results with $p_{offset}=75$ and $p_{offset}=10$, $p_{offset}=75$ shows better results in terms of *min(h)* and *min(f)*. The minimum values for a $p_{offset}=10$ refer to the identical solution. However, the maximum value solutions of $p_{offset}=75$ have larger values than those of $p_{offset}=10$. In addition, the average values of $p_{offset}=10$ are smaller than those of $p_{offset}=75$, except for average cost. The standard deviations of $p_{offset}=75$ are much larger than those of $p_{offset}=10$.

Overall, method M2 with $p_{offset}=75$, under ideal conditions, can obtain better results. However, it also produces much worse solutions. Considering the average and standard deviations of the solutions, method M2 with $p_{offset}=10$ is more directed and solves the problem more reliably. Further, the penalty value is zero for all solutions obtained with $p_{offset}=10$. Therefore, $p_{offset}=10$ is used as penalty offset in subsequent studies.

6.2.3 Simulated Annealing for outer loop search

To provide a comparison to A* for the outer loop search, Simulated Annealing (SA) is applied. Figure 6-6 shows solutions of Simulated Annealing (SA) and Pattern Search using penalties (M3) for the example part comparing to the method M2. However, only a few solutions were obtained for method M3. The reasons are the long calculation time comparing to A* and hang ups of the implementation during Boolean operations that are required for the calculation of the objective function.

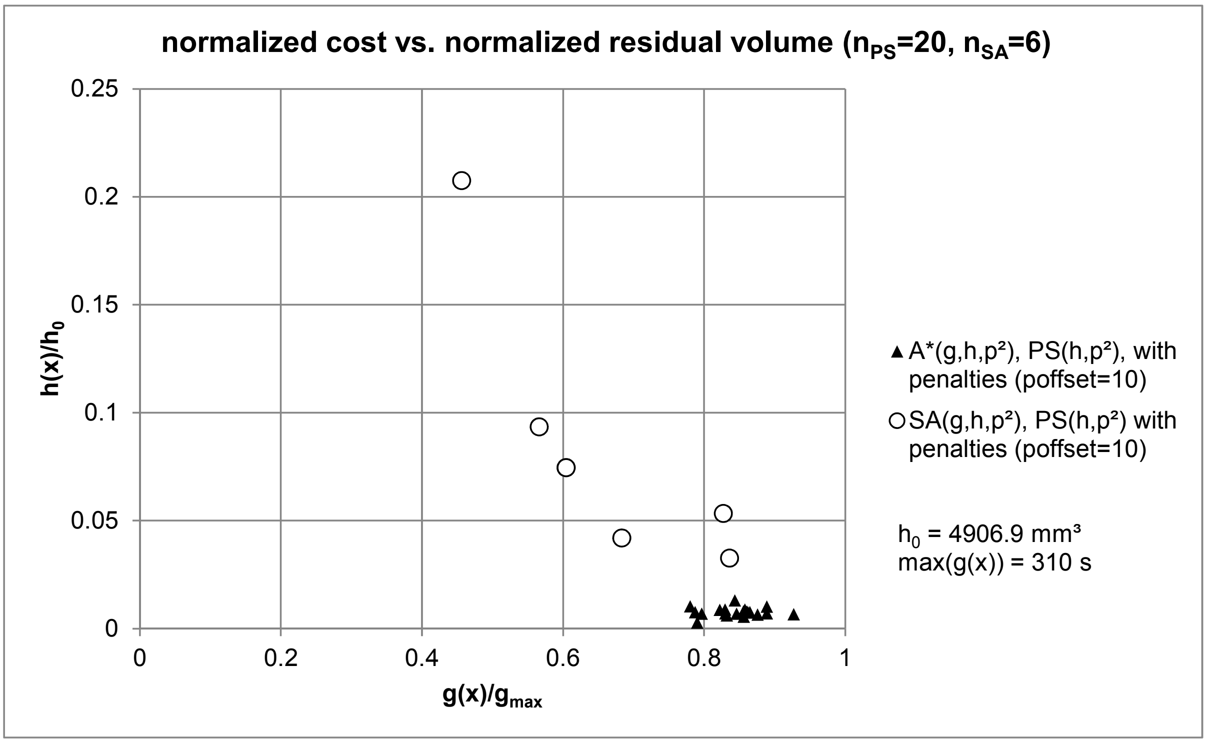


Figure 6-6: Approximated cost vs. normalized residual volume (error value) for A-Pattern Search with penalties(M2) and Simulated Annealing (SA) with Pattern Search and penalties (M3).*

Comparing the results, the solutions obtained by using M3 are worse with higher objective function values and higher residual volumes than those obtained by using M2 (A*). However, only a few numbers of results using SA were successfully generated. In most solution runs, the SA search did not finish but hung during calculating Boolean operations. This behavior is due to the more exploratory characteristic of SA during the early phase of the search process. Here, SA also accepts inferior solutions, whereas A* always accepts the best solution. If inferior solutions are accepted in the beginning, the search can be lead to areas of the search space where the objective function cannot be calculated anymore due to the lack of stability in the Boolean operations.

Table 6-5: Result statistics by methods, best and worst solutions for h and f, average values and standard deviations.

		best min(h)	best min(f)	worst max(h)	worst max(f)	average	standard deviation
A*-PS with penalties (M2, p_{offset}=10)	h [mm³]	13.5313	13.5313	63.5919	63.5919	37.4282	9.8656
	hres / h0	0.0028	0.0028	0.0130	0.0130	0.0076	0.0020
	g [s]	245.1050	245.1050	261.5800	261.5800	261.9818	11.3671
	p [mm³]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	f	129.3180	129.3180	162.5860	162.5860	149.7049	7.6878
SA-PS with penalties (M3, p_{offset}=10)	h [mm³]	160.2310	205.4680	1017.7700	1017.7700	404.9111	267.7536
	h_{res} / h_0	0.0327	0.0419	0.2074	0.2074	0.0825	0.0546
	g [s]	259.2480	211.8230	141.5490	141.5490	202.7660	39.8574
	p [mm³]	0.0000	0.0000	1.1538	1.1538	0.1648	0.4037
	f	209.7390	208.6460	590.9900	590.9900	305.4571	121.9149

6.2.4 Simulated Annealing for re-optimization

The search methods presented suffer from having only local information on the effect of variable changes. For example, the selection of variables, e.g. how long a slot cut is made, in the early search process can lead to inferior solutions in the late search process. This is due to the two-level search process that is necessary to optimize the topology of solutions, i.e. which machining operations, as well as the variables within that topology.

To overcome the limitation of the two level search process, the variables within the existing topology are optimized at the same time such that the objective function of the last resulting state (leaf) is minimized. Improvements are achievable by reducing overlap of Removal Volumes as well as to remove cusps in the shape of the resulting machining plan. Through this, cost *g(i)* as well as the residual volume *h(i)* can be reduced.

Figure 6-7 shows the principle of the re-optimization. On the left side, the state tree is displayed that stores all evaluated states that represent partial solutions. In the first step, the current best state and all of its predecessors are copied to a separate branch. This allows manipulating the states while not violating the integrity of states that are adjacent to the active branch (white circles).

After copying the branch, the optimization can commence. During optimization, neighboring solutions are generated and evaluated. The evaluation comprises the updating of all states in the optimization branch from top node to leaf node. The objective function for the optimization is then the objective function *f(i)* of the last state of the optimization branch. After re-optimization, the regular search process can continue with the new best state.

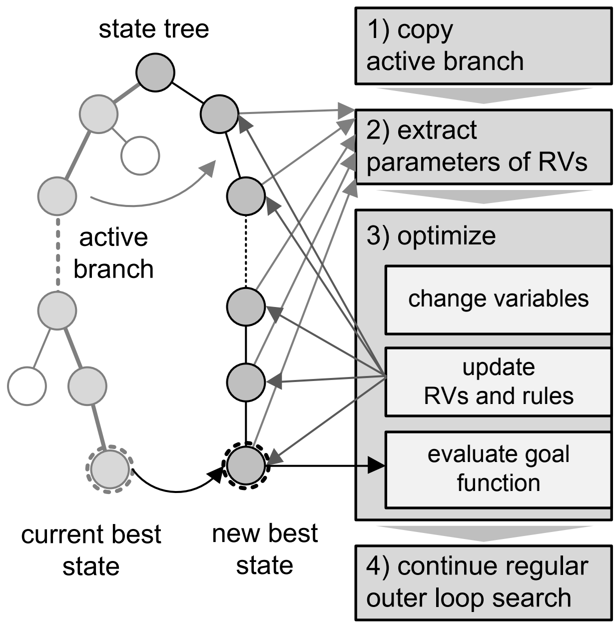


Figure 6-7: Re-optimization of a solution

The generation of neighboring solutions is crucial in this closely coupled optimization problem. Since the location of a RV is defined by the end-point of the preceding tool-path, any change of variables also results in changing shapes of all succeeding RemovalVolumes. Figure 6-8 displays this situation. The first of the three RVs at the bottom is changed by moving the end-point of the RV upwards denoted by the arrow. The orientation of the toolpath is counter-clockwise, as denoted by the triangles.

If the size of the RV and the associated toolpath is changed, the succeeding RVs are updated and change their positions to maintain an uninterrupted tool-path. This, however, can result in the succeeding RVs to be outside of the TRV. This makes optimizing the variables almost impossible since a change of variables can result in a drastic increase of the objective function due to high penalty values *p(i)*.

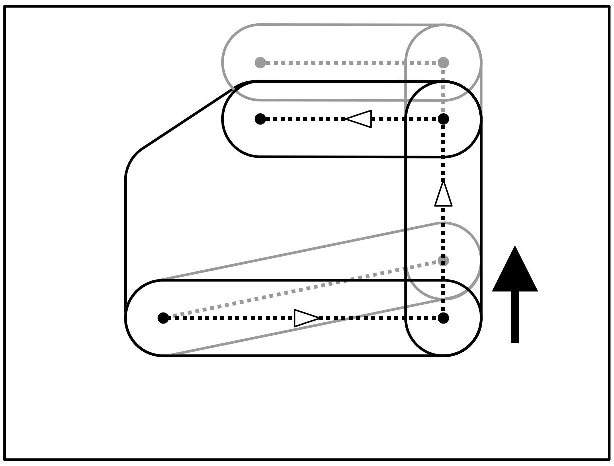


Figure 6-8: Influence of variable changes on succeeding RVs

To overcome this, only one RV should be changed in each iteration and the succeeding RVs updated to have the same end-point of the new toolpath.

This method is shown in Figure 6-9. Here, the RV at the bottom is changed. Once the parameters of the RV are changed, the following, vertical RV is changed too, to have the same end-point as before. Through this, the changes are more directed and cause less penalties during search.

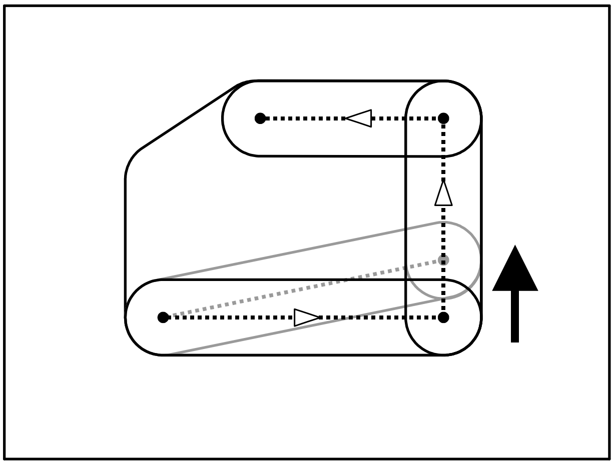
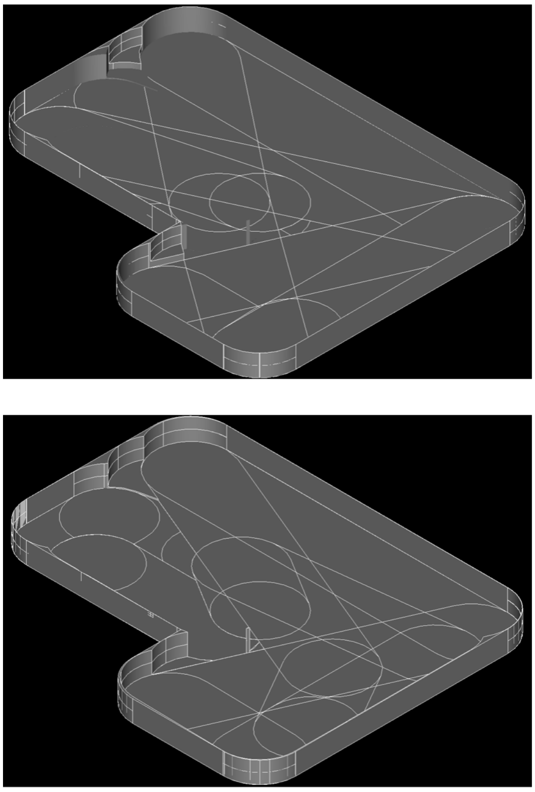


Figure 6-9: Generating neighboring solutions during re-optimization

One solution, obtained by using A* and Pattern Search to create a solution candidate and then using Simulated Annealing for re-optimization, as described, is shown in Figure 6-10. The solutions candidate before re-optimization is shown at the top (a). The final solution after re-optimization is shown at the bottom (b). After the re-optimization the residual volume is higher whereas the cost is lower than before. Also the objective function f is lower. Comparing the shapes, before, there are two partial cuts leaving triangle shaped plates at the bottom of the TRV. After re-optimization, these areas have been removed but another cusp has been created in the right corner. However, the search time of 72072s to achieve this result is large compared to the time of 1780s, needed to create the initial solution candidate.



a

b

Figure 6-10: Graphical representation of the machining plan before (a) (h=40.9934, g=290.91, p=0, f=165.952 computation time t=1780s) and after (b) (h=56.5631, g=243.084, p=0, f=149.824, computation time t=72072s) re-optimization (M4).

The behavior of SA during the re-optimization will now be explored to ensure that the behavior is not the result of a flaw within the implementation. Figure 6-11 shows the acceptance rate and the acceptance target rate over the iterations of the Simulated Annealing search run. The acceptance target follows the Lam-Delosme-schedule. The actual acceptance rate follows the target values rather well during the first 65% of the iterations. Afterwards, a larger difference between the target value and actual value occurs. Especially after 85 iterations, the actual acceptance rate is almost constant whereas the target value is lowered further.

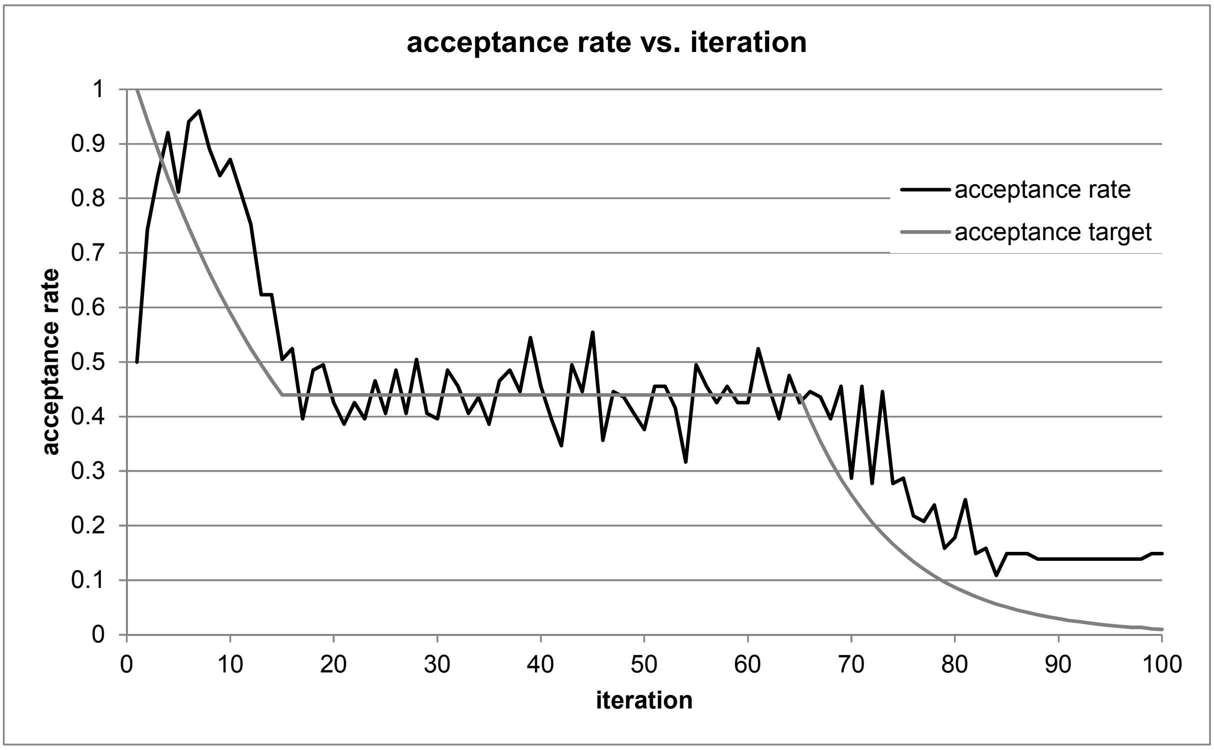


Figure 6-11: Acceptance rate and acceptance target rate.

Figure 6-12 shows the course of the SA temperature over the iterations of the search. The temperature is the control parameter of the acceptance rate. During the first 15 iterations, the temperature is kept high. This phase is called initial quench. The high temperature should allow the search to leave local optima. During the simmer phase until iteration 65, new optima are explored. Towards the end of the search, the temperature is lowered until the search freezes and ideally converges towards optima. However in this particular search run, the minimum temperature of 0.001 is reached already at iteration 85.

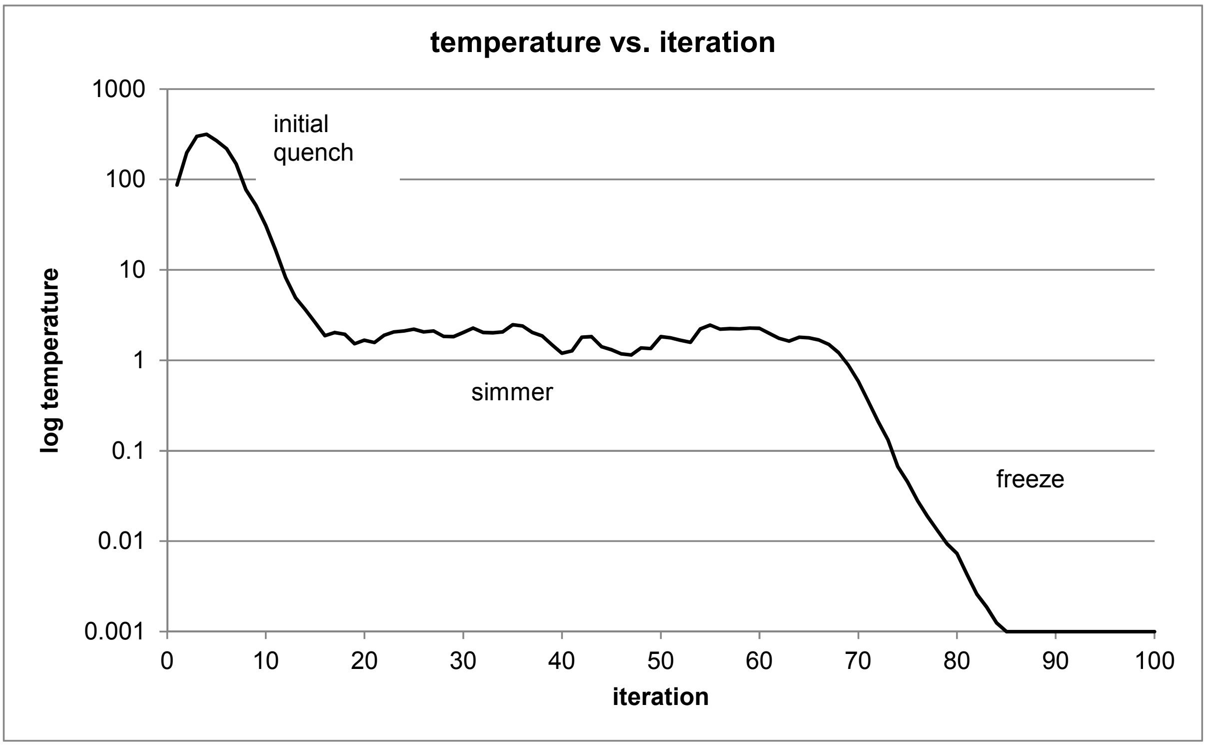


Figure 6-12: Simulated Annealing temperature vs. iterations.

To study the behavior further, overall 104 successful machining planning runs using A* and SA re-optimization were run. The number of evaluated states during A* search is lowered to 500 and the number of Simulated Annealing re-optimization iterations and moves are each set to 100. However, a validation of results showed numerical problems in calculating the objective functions. A validation of very good results, i.e. low residual volume *h* below 2mm³, showed that these are due to numerical inaccuracies during volume calculation caused by inverted faces. Also numerical aberrations in calculating the objective values occurred. These caused lower residual volume *h* while maintaining the same plan and cost *g*. Since the plan is identical, the residual volume cannot change. Therefore, these two cases were removed from the result set.

Figure 6-13 shows the normalized residual volume and normalized cost for the results using A* outer search only (M2) and using A* and Simulated Annealing re-optimization (M4).

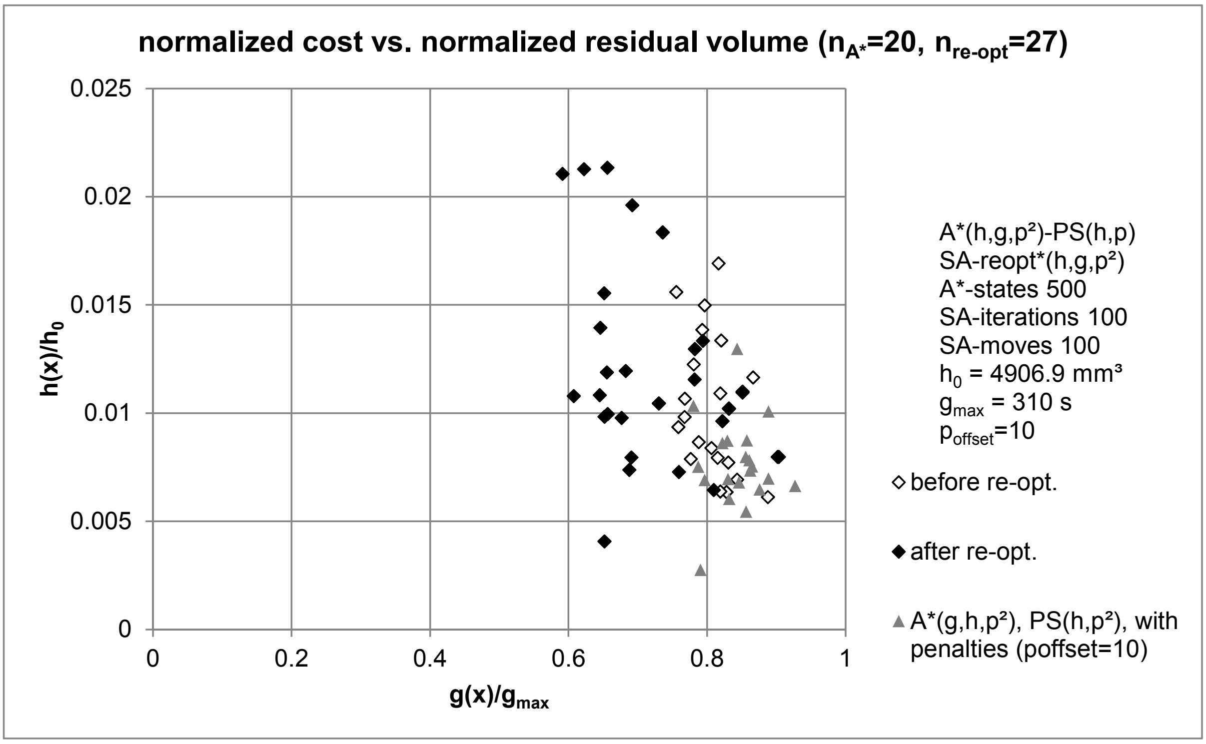


Figure 6-13: Approximated cost vs. normalized residual volume (error value) for A-Pattern Search with penalties (M2) and A*-Pattern Search with penalties and Simulated Annealing (SA) re-optimization (M4).*

Comparing the results using A* only (M2) and A* with re-optimization (M4), M2 only shows better results than M4 before the re-optimization run. However, this behavior was expected due to the lower number of evaluated states in the A* search (500 compared to 1000). After re-optimization, the results tend to have lower cost while also exhibiting larger residual volumes.

The results obtained using Simulated Annealing re-optimization can be differentiated as follows:

- Case 1: no change in the objective function, i.e. the search is unable to find a better result (4 occurrences)
- Case 2: lower objective function values *f* and lower values for residual volume *h* and cost *g* (4 occurrences)

- Case 3: lower objective function value *f* but higher value for residual volume *h* and lower cost *g* (15 occurrences)
- Case 4: lower objective function value *f*, lower residual volume *h* but higher cost *g* (2 occurrences)
- Case 5: marginal changes in objective function values *f*, *h* and *g* (less than 0.1) (2 occurrences)

Table 6-6 comprises the detailed statistics for each method.

Table 6-6: Result statistics by methods, best and worst solutions for h and f, average values and standard deviations.

		best min(h)	best min(f)	worst max(h)	worst max(f)	average	standard deviation
method M2 (n=20)	h [mm³]	13.5313	13.5313	63.5919	63.5919	37.4282	9.8656
	h_{res} / h_0	0.0028	0.0028	0.0130	0.0130	0.0076	0.0020
	g [s]	245.1050	245.1050	261.5800	261.5800	261.9818	11.3671
	p [mm³]	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	f	129.3180	129.3180	162.5860	162.5860	149.7049	7.6878
	comp. time [s]	2916	2916	3067	3067	3106	111
method M4 (n=27)	h [mm³]	19.9284	52.8789	104.7010	96.1619	59.2691	22.5740
	h_{res} / h_0	0.0041	0.0108	0.0213	0.0196	0.0121	0.0046
	g [s]	202.1380	188.4600	203.5520	214.6370	221.2876	25.1113
	p [mm³]	0.6402	0.0000	0.0000	1.3091	0.5867	0.6511
	f	121.4430	120.6700	154.1270	167.1130	145.8613	11.5550
	comp. time [s]	42502	61607	49211	46322	51170	5851

Comparing the two methods, M2 and M4, M2 clearly shows better results for minimum residual volume. In all cases and on average, the residual volume obtained using method M2 is lower. However, cost *g* is significantly lower using method M4. This leads to lower values of the objective function *f*, except for the worst result with maximum *f* value using method M4. Here, the objective function *f* is higher. This is also reflected by the higher standard deviation of *f* of method M4. On average, M2 reaches lower values in residual volume *h* and penalty volume *p* with lower standard deviations. However, on average, M4 reaches lower values in cost g and objective function f but with higher standard deviations. Overall, M4 reaches better results for the objective function f but the computation time to obtain these results is on average a factor 16 times higher than using method M2.

6.3 Discussion

Advanced heuristic search methods have been introduced that further improve the search. The size of the penalties has an influence on the result quality. With a lower penalty offset it is possible to obtain more directed solutions to the problem using A* and Pattern Search with penalties.

The simulated annealing algorithm using Hustin moves allows for adapting the rule application policy. Through this, effective rules are applied more frequently than less effective ones. However, Simulated Annealing is known to be very exploratory considering the design space and sometimes also exploits the model it is applied to. It turned out, that SA often does not finish the search due to numerical problems in calculating the objective function. Through this, only inferior solutions are returned while better solutions are not found since the search stops at some point.

In most cases, the best results are not yet satisfying since residual volume is still left over. To obtain a perfect shape match between the desired part and the fabricated part, the residual volume should be zero. To overcome this, re-optimization techniques were introduced. Motivated by the local knowledge during rule application, i.e. the rule is applied only considering the current and not any future state, all applied rules are optimized in one search. Results show that Simulated Annealing is able to further improve the results in terms of the objective function due to its ability to escape local minima and accept inferior moves in the early stages of the search. However, the improvement of the objective function value can also cause an increase of the residual volume that is offset by a greater decrease in fabrication time.

6.4 Conclusion

Three different heuristic search methods and their validation have been presented: A* search, Simulated Annealing with Hustin moves and a combination of A* with re-optimization of the best result using Simulated Annealing.

The results indicate, that A* as well as A* and Simulated Annealing re-optimization solve the problem successfully. However, the computational effort for re-optimization is high compared to the benefit gained. The high computational effort results from the complex objective function evaluation due to the recursive calculation of cost and penalty. Also, minimizing the objective function *f*, basically the sum of residual volume and cost of the solution, can lead to increased residual volume. This of course results in a larger deviation of the fabricated part from the designed shape, which might not be acceptable.

Although Simulated Annealing with Hustin moves seemed promising, the application of this search method did not produce better results. Due to the exploratory nature of this stochastic search algorithm, the representation and manipulation algorithms underlying the planning system become unstable and are not able to successfully compute solutions. Although it is possible to obtain some results, it is clear that these do not successfully solve the problem but only represent cases were the search did not try too hard to find good solutions and therefore did not make the representation for the search fail.

7 Discussion

In the following the results presented in this thesis are discussed and the contributions to the state of the art are reviewed, followed by a discussion of the current limitations of the method and its implementation. The thesis is closed by recommendations for future work and a final conclusion.

Machining planning deals with the arrangement and spatial relation between geometric shapes and the fabrication operations that create them. Spatial grammars natively support such a geometry-based planning by their ability to represent and manipulate spatial relations and entities. By applying elements of the vocabulary representing the geometric shape created by a machining operation, and using rules under constrained spatial relations, a feasible process plan can be created. The combined rules and vocabulary represent the fundamental knowledge of the machining process. Implicitly, the set of manufacturable shapes is represented by the grammar as well.

The method combines multiple planning steps together instead of using a hierarchical approach that is partitioned into manufacturing volume recognition, toolpath planning and scheduling. The main advantage is the use of “deep” knowledge of machining processes encoded in the rule set and vocabulary. Thus, fabrication constraints are already integrated in the planning process, however, in a generic way. Starting from a 3D shape of the part, the approach is able to carry out all planning tasks including toolpath planning in a bottom-up fashion. The ability of the method to use a dynamic rule set and vocabulary in the grammar, by (un)registering resources that correspond to machining capabilities and tools and allows for responding to changes in the hardware due to failure or (un-)availability. This capability can also be used for extending the approach to different manufacturing processes such as turning in addition to 3-axis milling.

To extend the method, new Removal Volumes can be created and used by the existing rules, if the same constraints apply, or if necessary, additional generic rules can be created specific to other cutting processes. The use of generic rules combined with search to direct the application of rules allows for easier extension to a wider range of machining strategies and machining processes compared to putting all the knowledge in the rules themselves. This also reduces the negative impact of rule maintenance. Conventional 2.5D default strategies, e.g. creating toolpaths by offsetting the part contour or machining the part in a line-by-line fashion, can result in effective process plans but are limited to their specific application area and workpieces. The use of the dynamic grammar overcomes the necessity to model all knowledge upfront. Further, it avoids maintaining a feature library and separate mappings from features to specific manufacturing operations.

The machining planning system can also be used to have a design-to-fabrication system run autonomously, i.e. unattended. By reacting to changes in the availability of workpieces and reacting to tool failures the system can overcome critical situations that would cause a traditional manufacturing system to shut down. In the case of reacting to the unavailability of workpieces in traditional CAM, it would require a complete manual re-planning of the process. The envisaged system, based around the SGMP method, however, decides if it can overcome the problem of the missing workpiece.

During the development of the method, several search methods have been applied: A* with Pattern Search under hard constraints, A* with Pattern Search and penalties, Simulated Annealing with Pattern Search and A* with Pattern Search and Simulated Annealing re-optimization. A* with Pattern Search under hard constraints has been successfully applied to the planning problem. However, it turned out that the search gets stuck within certain situations due to the hard constraints that keep possible solution candidates from being accepted at all. To overcome this, the hard constraints were relaxed and transformed into a penalty that was introduced to the objective function. Through this, the search is able to solve more complex problems better than the method with hard constraints. Still, the results indicate room for further improvement. One issue with the deterministic nature of the A* algorithm is that it only accepts the next best solution and does not accept inferior solutions that might lead to the global optimum in the later search. To overcome this, Simulated Annealing, a widely known stochastic search algorithm was applied. However, it turned out that only few results can be obtained and the ones obtained are not satisfying in terms of the objective function. The majority of searches did hang due to numerical problems within the solid model representation and the Boolean operations. This once more, proves the power of Simulated Annealing being able to explore the model it is applied to such that the model fails. Since the results were not improved by using Simulated Annealing, re-optimization techniques were introduced. In contrast to the previous search methods, re-optimization does not apply a single rule or RV but attempts to alter a complete process plan at once. This technique allows to consider knowledge on the effect of rule applications, e.g. if the first rule is changed, its effect on all subsequent rules is considered in the re-optimization. Therefore the search is able to not only consider the next rule application but to consider all. The results from applying re-optimization prove its feasibility but the computation time is extremely high compared to all other search methods. In conclusion, A* in combination with Pattern Search and penalties, despite its simplicity, still offers the best tradeoff between computation time and quality of the solutions for this application.

In analogy to the analysis of using a commercially available CAD/CAM software tool, the different steps using the developed method and software prototype are shown in Figure 7-1. The process begins by defining the part geometry by importing a 3D part model. The origin is already pre-defined by the geometric model. The stock can be defined either by importing a stock geometry from 3D model or by using the workpiece selection system presented in SHEA et al. 2010. By using a metric for selecting a specific workpiece, this step can also be automated. Since the available hardware system and the automated handling devices are constrained to a single setup, the setup is already pre-defined. However, the workpiece could be machined on several different machine tools with the same setup orientation.

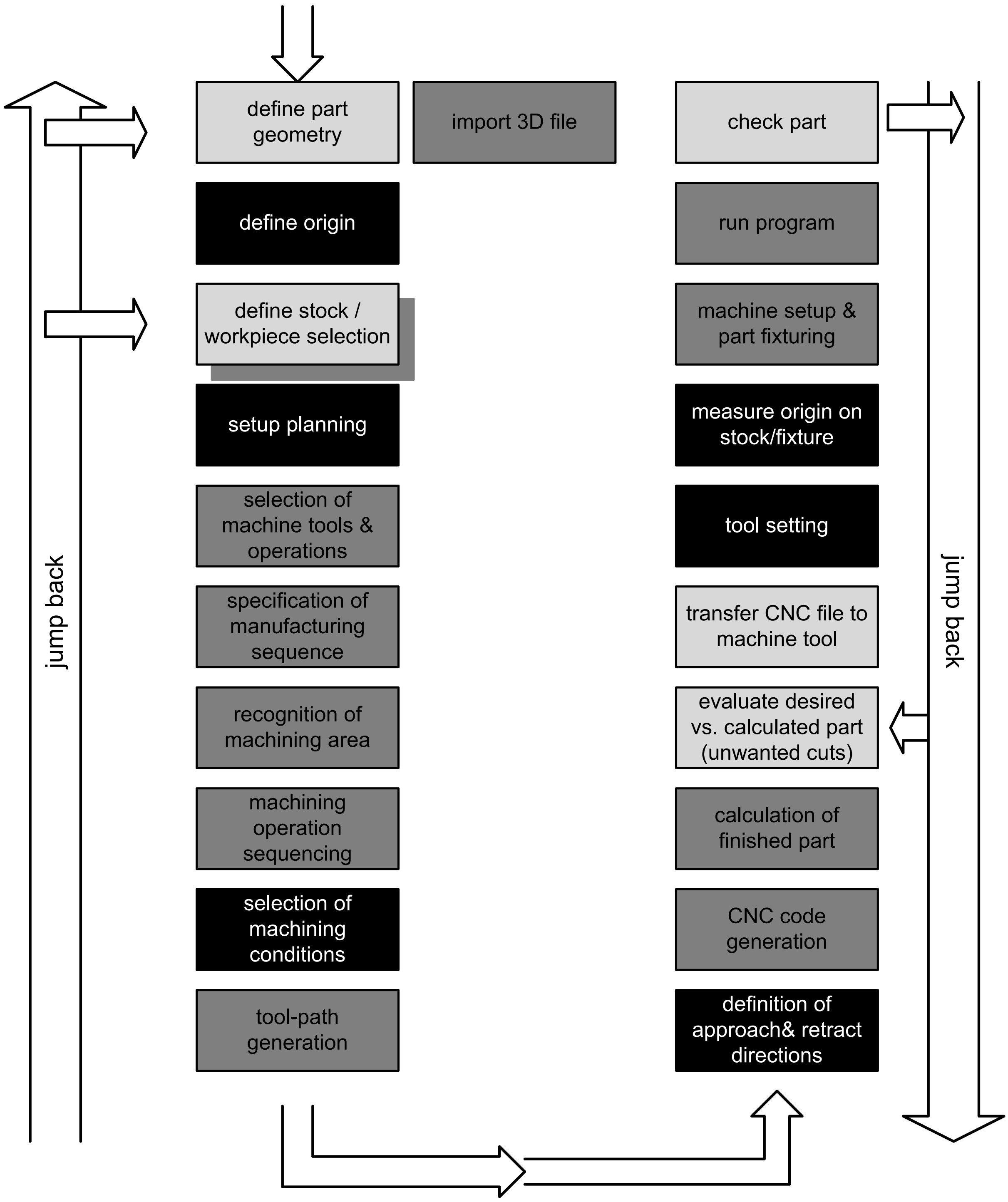


Figure 7-1: Study of process planning using Spatial Grammar Machining Planning (SGMP). Different shades of color indicate the degree of automation from human based (light gray) to machine based (dark gray) and pre-defined or constrained by the available hardware (black).

Within the heuristic search process, the machine tools and operations as well as the sequence of manufacturing operations are determined automatically. Further the machining area is recognized and the operations sequenced automatically. The machining conditions (i.e. speeds and feeds) are pre-defined. Using the available data, the tool-paths can then be generated. Since there is only a single setup due to the constraints of the systems, the approach and retract directions are already defined. However, the exact trajectory is still calculated automatically. After this the CNC code can be generated. Since the system capitalizes on spatial grammars to integrate the shaping capability of the machine tools, the finished part is automatically calculated throughout the machining planning process.

After the user has evaluated the calculated finished part, the CNC code can be transferred to the machine tool. The file transfer can be automated. Since the Spatial Grammar Machining Planning (SGMP) already plans with the tools available at planning time, the tool setting is already defined upfront. Since the machine tool is fed automatically by a handling device, the origin on the fixture is pre-defined. However, the part fixturing is carried out automatically by a reconfigurable fixture presented in SHEA et al. 2010. After the program has run, the part has to be checked manually.

7.1 Review of research contributions

A new bottom-up process-centric approach for CNC machining planning for the fabrication of customized parts for autonomous design-to-fabrication systems has been presented. The approach combines methods from design synthesis, namely spatial grammars and heuristic search methods that are used to generate designs and to decompose a design into machining operations and CNC machining instructions. The achievements realized in this work are shown in Figure 7-2. In the area of models, for the first time a spatial grammar formalism has been used to capture the shaping capability of machine tools in a bottom-up fashion such that the current state of the machine tool and its capabilities can be reflected to the planning. Though the model has been applied to 2.5D milling in this work, it is not limited by design to this application. Further applications are the planning of turning and multi-axis milling processes. To address this machining process, new rules and vocabulary of shapes need to be designed that capture the shaping capability as well as the inherent process constraints.

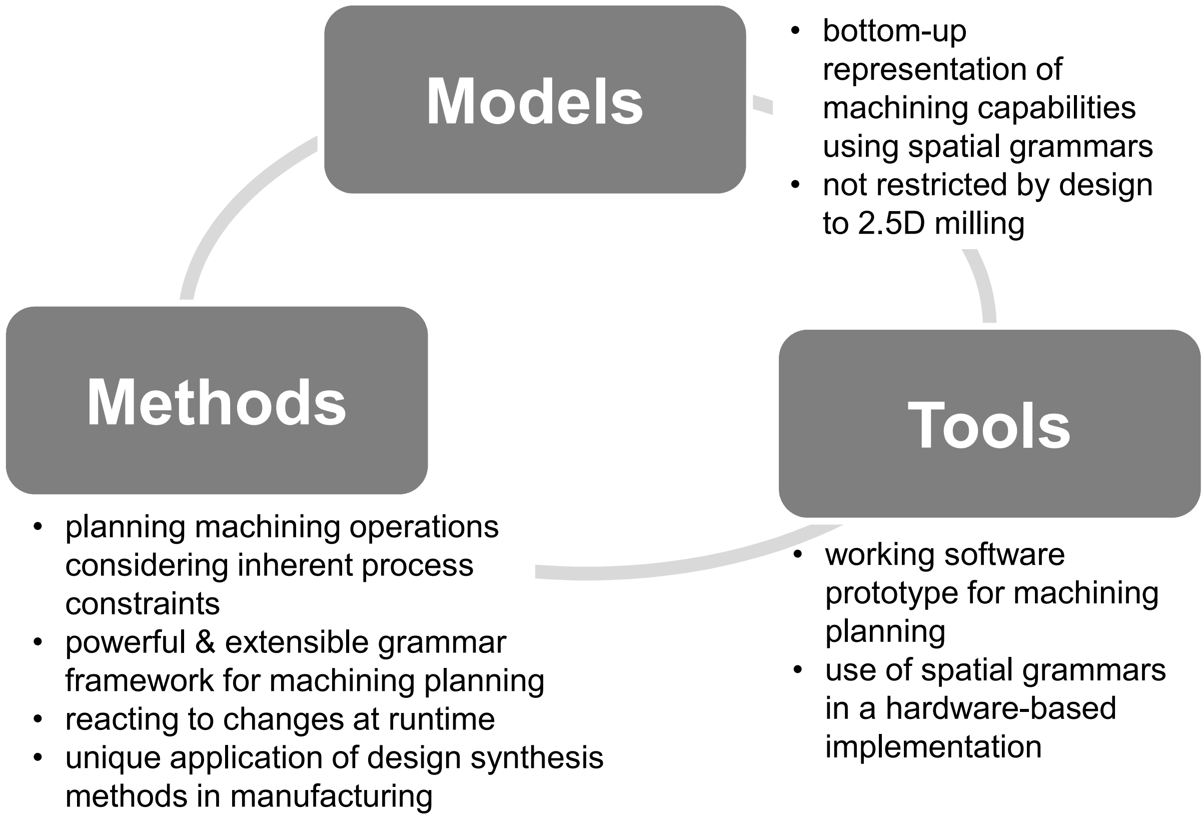


Figure 7-2: Research contributions in the areas of models, methods and tools.

In the area of methods, a method for the planning of machining operations using the previously described model has been developed. In the first version, it was limited to design specifications using shape grammars. An extension of the method to accepting general 3D solid models as input for the designed part has been presented as the focus of the thesis. With this method, a customized design and the workpiece can be specified using CAD software. From this representation, the method is able to generate the machining instructions to fabricate the part from a specified workpiece. The feasibility of the approach has been demonstrated by successfully planning the machining operations and executing these on the hardware to fabricate the part. The powerful and extensible grammar framework allows for adding and changing capabilities. Through this, the system can react to changes in the hardware online. If workpieces are not available or a tool failure has been detected, the system can still successfully fabricate the desired part. The current state of the fabrication system is reflected in the planning system. The ability to incorporate such online feedback has been demonstrated in the scenarios handling missing workpieces and reacting to tool failure. The method and its implementation are unique within design synthesis methods, and specifically spatial grammars, in manufacturing to decompose a design into feasible machining operations that are linked to available hardware and capabilities.

The model and method has been implemented using a geometric kernel for solid modeling to support the capabilities of a three-dimensional set grammar. A working software prototype for machining planning for 2.5D milling processes has been implemented. The application allows importing STEP solid models of the part and workpiece, inspection of the resulting Total Removal Volumes and the choice of different search methods to generate the machining plan. By having the generated machining plans executed on CNC machine tools, the hardware-based implementation demonstrates the integration of spatial grammars with hardware for fabrication.

Overall the approach is an enabler for autonomous design-to-fabrication systems. Through reacting to feedback from the hardware and online planning capabilities, the actual planning process can be pushed downstream to the machine tools. Through this, the machine tools can be made aware of their capabilities and enabled to reason about part geometry in relation to available capabilities and to plan on-line their own operations to fabricate a part. The machine tools can not only plan their actions but also respond robustly to changes and unforeseen events within the design-to-fabrication system and maintain operations by overcoming failure of resources and machines.

7.2 Limitations

One of the limitations of the current implementation is the limited part complexity that the system can handle and successfully generate machining plans for. One reason for this limitation is the numerical complexity of the Boolean operations that are used for the calculation of the objective function as well as for the simulation of the individual cutting operations. During search, the calculation of the Boolean operations is very time consuming such that the search hangs, i.e. even after several hours, when a single Boolean operation does not completely compute. Another reason is the combinatorial explosion if further rules and Removal Volumes are added to the system. Depending on the search method applied, all available rules and Removal Volumes must be tested before a choice for a specific rule or Removal Volume is made. With a large branching factor, i.e. a large number of rules and RVs, the speed of the search progress decreases.

The slow search progress is another limitation that makes the application of the system online more difficult. The machining plan generation can take one to several hours even for a rather simplistic part. The primary reason, besides the already mentioned ones, is the rather slow representation of the geometry using solid modeling. While being the first choice considering exactness, the computation of the geometry is very time-consuming. Here alternative and approximated 3D geometry representations could help increase the speed of the search progress.

Currently the Removal Volume generation is not implemented. Instead, the Removal Volumes need to be preprogrammed. However, the Removal Volume generation requires complex spatial operations. Therefore it was decided to not implement the Removal Volume generation to not further slowdown the search.

7.3 Recommendations for future work

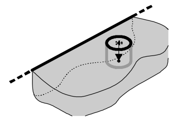
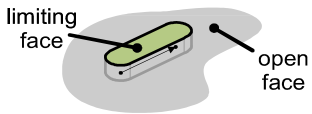
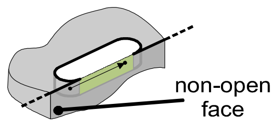
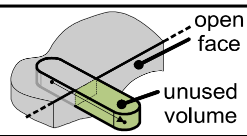
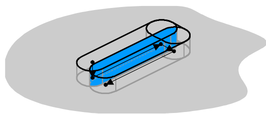
For the further development of the method and implementation the primary limitations should be addressed. The root cause for several limitations lies in the representation of the 3D geometries. The Boolean operations are slow in comparison to the rest of the search and, more important, are prone to fail, especially when the solution becomes more accurate, i.e. with low residual volume. Therefore, alternative representations for 3D geometries should be explored. Examples are octree or volumetric mesh representations. These are only approximate representations but are much faster to compute. It would also be possible to combine the advantages of approximate representations with solid modeling techniques. By using the approximate representation to generate and test solutions candidates and the solid model then to calculate the final solution, the search can be made faster with a similar level of exactness.

With a faster geometric representation, it would be possible to implement and use the Removal Volume generation online. Through this, the machine tools can take advantage of newly added tools instantly after installation. With the use of tool shape scanning methods, available in today's machine tools, the tool shape can be captured and used in the planning of the machining operations. The automated Removal Volume generation would also ease the extension of the implementation to alternative machining processes such as turning. By specifying the tool shape and the kinematic properties of the process it would be possible to capture the knowledge of turning processes.

Addressing the increasing complexity of plan generation with growing capabilities, or to be more exact with increasing number of rules and Removal Volumes, improved heuristics and soft-constraints are needed to constrain the search to promising areas of the search space. These heuristics and constraints should integrate expert best practice knowledge in the planning process, such that the system can generate better machining plans and achieve higher quality of fabricated parts.

A number of such soft constraints are shown in Table 7-1. These soft constraints are classified according to their primary intent: Whether they reduce the complexity of the planning process or whether they are used as planning strategy to create better quality machining plans. It is important to mention that the soft constraints may be violated and still a feasible plan can be generated, i.e. they provide guidance to the search and incorporate additional domain specific knowledge. One or multiple constraints can be incorporated as penalties in the objective function during the search as described in Section 4.

Table 7-1: Soft constraints on spatial relations for rule application

name	location	spatial relation type	spatial relation	contribution to		example
				planning strategy	reducing complexity	
C.7	R.1	RV:TRV	toolpath origin on open face of TRV		X	
C.8	R.3	RV:TRV	Coincident open faces and limiting face should give good results	X		
C.9	R.3	RV:TRV	Coincident non-open face and a non-limiting face should be good	X		
C.10	R.3	RV:TRV	Open faces may be penetrated by the RVs but can be less effective	X		
C.11	R.3	RV:RV	To reduce cusping, RVs should overlap	X		

C.7 recommends that an initial toolpath should start from an open face of the TRV. Similarly to this, **C.8** recommends having coincident limiting faces of the RV and open faces of the TRV thus using the full volume of the RV for the machining operation. To produce a good surface quality, **C.9** recommends having coincident non-limiting faces of the RV and non-open faces of the TRV such that the surfaces each are created by as few as possible RVs.

C.10 aims at minimizing unused volume of the RVs which results in longer machining time. This is a soft constraint since it is not required but gives a hint about how to create an efficient process plan.

C.11 aims at increasing the machining quality by having the individual RV overlap thus reducing cusping in the machining process.

An example for a heuristic to improve the search is the implementation of an improved repositioning rule. Instead of repositioning the tool randomly, it could be placed above an area where the most residual volume exists.

7.4 Conclusion

The feasibility of using spatial grammars in combination with heuristic search for CNC machining planning from CAD geometries has been demonstrated by the results presented. The approach and the implemented Spatial Grammar Machining Planning (SGMP) system incorporate machining knowledge directly in the planning. This is achieved by the representation of machining operations through the spatial grammar. Since the shape is directly associated with the machining operation, there is no need for mapping the geometry separately to the machining operation as in feature technology. The availability of machines and tools can be reflected by the SGMP system by activating and deactivating the vocabulary and rules that the machines and tools are associated with. Through this, newly added tools can be used directly without mapping to machining operations and the unavailability of resources can be considered in the planning. Since the altering of the rule set is possible during planning, the planning system can cope with highly dynamic situations within a fabrication environment. Therefore, the method can be used for online planning of fabrication processes. The SGMP system is able to generate a full fabrication plan from operation selection and sequencing, tool and resource selection down to the generation of specific CNC code and therefore integrates CAPP and CAM capabilities without the need of post-processing data for the fabrication of parts. Thus, the approach enables direct fabrication of customized parts on CNC machines. Through this, the spatial grammar provides knowledge about the shaping capability of the machine tools and the SGMP systems allows the machine tool to plan its own actions. Therefore, the approach is an enabler in the development of cognitive machine tools within the Cognitive Machine Shop that "know what they are doing" (BRACHMAN 2002).

8 References

AGARWAL & CAGAN 2001

Agarwal, M.; Cagan, J.: On the use of shape grammars as expert systems for geometry-based engineering design. Artificial intelligence for engineering design, analysis and manufacturing AI EDAM 14 (2001) 5, p. 431-439.

AGARWAL et al. 2000

Agarwal, M.; Cagan, J.; Stiny, G.: A micro language: generating MEMS resonators by using a coupled form - function shape grammar. Environment and Planning B: Planning and Design 27 (2000) 4, p. 615-626.

AHN et al. 2001

Ahn, S. H.; Sundararajan, V.; Smith, C.; Kannan, B.; D'Souza, R.; Sun, G.; Mohole, A.; Wright, P. K.; Kim, J.; McMains, S.; Smith, J.; Sequin, C. H.: CyberCut: An Internet-based CAD/CAM System. Journal of Computing and Information Science in Engineering 1 (2001) 1, p. 52-59.

ALBERT 1990

Albert, M.: Interfacing NC with CIM. Modern Machine Shop 63 (1990) 2, p. 66-81.

ALEXANDRESCU 2001

Alexandrescu, A.: Modern C++ design: generic programming and design patterns applied. Boston, USA: Addison-Wesley 2001. ISBN: 0201704315. (C++ in depth series).

AMAITIK & KILIÇ 2007

Amaitik, S. M.; Kiliç, S. E.: An intelligent process planning system for prismatic parts using STEP features. International Journal of Advanced Manufacturing Technology 31 (2007) 9, p. 978-993.

ARAS & YIP-HOI 2008

Aras, E.; Yip-Hoi, D.: A Solid Modeler Approach for Extracting Cutter Workpiece Engagements in Milling, Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE. Brooklyn, New York, USA, August 3-6 2008.

ARKIN et al. 2000

Arkin, E. M.; Fekete, S. P.; Mitchell, J. S. B.: Approximation algorithms for lawn mowing and milling. Computational Geometry 17 (2000) 1-2, p. 25-50.

ARKIN & HASSIN 1994

Arkin, E. M.; Hassin, R.: Approximation algorithms for the geometric covering salesman problem. Discrete Applied Mathematics 55 (1994) 3, p. 197-218.

BEETZ et al. 2007

Beetz, M.; Buss, M.; Wollherr, D.: Cognitive Technical Systems - What Is the Role of Artificial Intelligence. In: Hertzberg, J. et al. (Eds.): Proceedings of the 30th German Conference on Artificial Intelligence, 2007.

BERNARD & KARUNAKARAN 2008

Bernard, A.; Karunakaran, K. P.: Rapid manufacturing of metallic objects: A challenge for research and industry. In: Bártolo, P. J. (Ed.) Virtual and rapid manufacturing, Proceedings of the 3rd International Conference on Advanced Research in Virtual and Rapid Prototyping, Leiria, Portugal, 24-29 September 2007 2008. Taylor & Francis ISBN: 978-0-415-41602-3.

BESANT & LUI 1986

Besant, C. B.; Lui, C. W. K.: Computer-Aided Design and Manufacture. 3rd ed. Chichester, UK: Ellis Horwood Limited 1986. ISBN: 0-85312-909-6. (Ellis Horwood Series in Mechanical Engineering).

BOURNE et al. 2011

Bourne, D.; Corney, J.; Gupta, S. K.: Recent Advances and Future Challenges in Automated Manufacturing Planning. Journal of Computing and Information Science in Engineering 11 (2011) 2, p. 021006-10.

BRACHMAN 2002

Brachman, R. J.: Systems that know what they're doing. Intelligent Systems, IEEE 17 (2002) 6, p. 67-71.

BRECHER et al. 2008

Brecher, C.; Kempf, T.; Herfs, W.: Cognitive Control Technology for a Self-Optimizing Robot Based Assembly Cell, Proceedings of the ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE 2008. Brooklyn, New York, August 3-6 2008.

BROOKS 1999

Brooks, R. A.: Cambrian Intelligence - The Early History of the New AI. Cambridge, Massachusetts: Bradford 1999. ISBN: 0-262-02468-3.

BROWN & CAGAN 1997

Brown, K. N.; Cagan, J.: Optimized process planning by generative simulated annealing. Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM) 11 (1997) p. 219-235.

BROWN et al. 1995

Brown, K. N.; McMahon, C. A.; Sims Williams, J. H.: Features, aka the semantics of a formal language of manufacturing. Research in Engineering Design 7 (1995) 3, p. 151-172.

BURNS 1993

Burns, M.: Automated Fabrication - Improving productivity in manufacturing. New Jersey: PTR Prentice Hall 1993.

CAGAN & MITCHELL 1993

Cagan, J.; Mitchell, W. J.: Optimally directed shape generation by shape annealing. Environment and Planning B: Planning and Design 20 (1993) 1, p. 5-12.

CHANG & WYSK 1985

Chang, T.-C.; Wysk, R. A.: An Introduction to Computer Aided Process Planning Systems. Englewood Cliffs, NJ, USA: Prentice Hall 1985. ISBN: 0-13-478140-6. (Industrial and Systems engineering).

CHANG et al. 1991

Chang, T.-C.; Wysk, R. A.; Wang, H.-P.: Computer-aided manufacturing. Englewood Cliffs: Prentice-Hall 1991. ISBN: 0-13-161571-8. (Industrial and Systems Engineering).

CHEN et al. 2006

Chen, Y.; Huang, Z.; Chen, L.; Wang, Q.: Parametric process planning based on feature parameters of parts. International Journal of Advanced Manufacturing Technology 28 (2006) 7-8, p. 727-736.

CORNEY et al. 2005

Corney, J.; Hayes, C.; Sundararajan, V.; Wright, P. K.: The CAD/CAM Interface: A 25-Year Retrospective. Journal of Computing and Information Science in Engineering 5 (2005) 3, p. 188-197.

COTESYS 2011

CoTeSys: Cluster of Excellence "Cognition for Technical Systems - CoTeSys" <http://www.cotesys.org> 2011-06-05.

CUTKOSKY et al. 1988

Cutkosky, M.; Tenenbaum, J.; Muller, D.: Features in process-based design. In: Tipnis, V. A. (Ed.) Proceedings of the 1988 ASME International Computers in Engineering Conference and Exhibition, San Francisco, California, USA, July 31 - August 4 1988. p. 557-562.

DESHAYES et al. 2006

Deshayes, L.; Welsch, L.; Donmez, A.; Ivester, R.; Gilsinn, D.; Rhorer, R.; Whitenton, E.; Potra, F.: Smart machining systems: issues and research trends. In: Brissaud, D. et al. (Eds.): Innovation in Life Cycle Engineering and Sustainable Development. Dordrecht, The Netherlands: Springer 2006, p. 363-380. ISBN: 978-1-4020-4601-8.

DIN 66025 1983

DIN 66025, part 1: Numerical control of machines, format; general requirements. Berlin: Beuth 1983.

DIN 66215 1974

DIN 66215, Part 1: Programming of numerically controlled machines; CLDATA, general structure and record types. Berlin: Beuth 1974.

DING & CAGAN 2003

Ding, Q.; Cagan, J.: Automated Trunk Packing with Extended Pattern Search, SAE 2003 World Congress & Exhibition. Detroit, MI, USA, 2003.

DRAGHICI & BONDREA 1998

Draghici, G.; Bondrea, I.: Integrated Approach in Computer Aided Process Planning, Proceedings of 1st International Symposium on Concurrent Enterprising. Sinaia, Romania, 1998.

FEENEY & FRECHETTE 2002

Feeney, A. B.; Frechette, S.: Testing STEP-NC Implementations. In: Proceedings of the 5th Biannual World Automation Congress, 9-13 June 2002 2002. p. 39- 44. ISBN: 1-889335-18-5.

FRANZ 1991

Franz, D.: CAD-CAM-Basisbetrachtungen. Berlin: Siemens Aktiengesellschaft 1991. ISBN: 3-8009-1578-2.

GEBHARDT 2003

Gebhardt, A.: Rapid Prototyping. Munich: Hanser 2003. ISBN: 3-446-21259-0.

GRABOWSKI & ANDERL 1990

Grabowski, H.; Anderl, R.: CAD Systems and Their Interface with CAM. In: Rembold, U. et al. (Eds.): Computer-Aided Design and Manufactruing - Methods and Tools. 2nd ed. Berlin: Springer 1990, p. 1-31. ISBN: 3-540-16321-2.

HATVANY 1982

Hatvany, J.: Internationaler Stand des CAD/CAM. In: Goebl, R. et al. (Eds.): CAD/CAM - Rechngestütztes Konstruieren und Entwerfen. Wien: R. Oldenbourg 1982, ISBN: 3-7029-0182-5. (Schriftenreihe der Österreichischen Computer Gesellschaft 16).

HEISSERMAN & WOODBURY 1993

Heisserman, J.; Woodbury, R.: Generating languages of solid models. In: Proceedings of the second ACM symposium on Solid modeling and applications, Montreal, Quebec, Canada, 1993. ACM p. 103-112. ISBN: 0-89791-584-4. (http://doi.acm.org/10.1145/164360.164400)

HELD 1991

Held, M.: On the Computational Geometry of Pocket Machining. Berlin: Springer 1991. ISBN: 3-540-54103-9. (Lecture Notes in Computer Science (LNCS)).

HOISL & SHEA 2011

Hoisl, F.; Shea, K.: An interactive, visual approach to developing and applying parametric three-dimensional spatial grammars. Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM) 25 (2011) p. 333-356.

HOOKE & JEEVES 1961

Hooke, R.; Jeeves, T.: "Direct Search" Solution of Numerical and Statistical Problems. Journal of the ACM (JACM) 8 (1961) 2, p. 212-229.

HUMMEL 1989

Hummel, K. E.: Coupling rule-based and object-oriented programming for the classification of machined features. In: Riley, D. R. et al. (Eds.): Proceedings of the 1989 ASME International Computers in Engineering Conference and Exposition, Anaheim, California, USA, July 30-August 3 1989. p. 409-418. ISBN: 0-7918-0357-0.

HUSTIN 1988

Hustin, S.: Tim: A new standard cell placement program based on the simulated annealing algorithm. Univ. of California, Berkeley, California (1988).

ISO 6983 2009

ISO 6983, Part 1: Automation systems and integration - Numerical control of machines - Program format and definitions of address words - Data format for positioning, line motion and contouring control systems. International Organization for Standardization 2009.

ISO 14649 2003

ISO 14649, Part 1: Industrial automation systems and integration - Physical device control; Data model for computerized numerical controllers - Part 1: Overview and fundamental principles. International Organization for Standardization 2003.

KEMPF et al. 2009

Kempf, T.; Herfs, W.; Brecher, C.: SOAR-based sequence control for a flexible assembly cell. In: Emerging Technologies & Factory Automation, 2009. ETFA 2009. IEEE Conference on, Palma de Mallorca, Spain, 2009. ISBN: 1946-0759.

KLAHR et al. 1987

Klahr, D.; Langley, P.; Neches, R. (Eds.): Production System Models of Learning and Development. Cambridge, Mass., USA: MIT Press 1987.

KOCHAN 1986

Kochan, D. (Ed.) CAM: developments in computer-integrated manufacturing. Berlin: Springer 1986. ISBN: 3-540-15165-6.

KOCHAN et al. 1999

Kochan, D.; Kai, C. C.; Zhaohui, D.: Rapid prototyping issues in the 21st century. Computers in Industry 39 (1999) 1, p. 3-10.

KRISHNA & RAO 2006

Krishna, A. G.; Rao, K. M.: Optimisation of operations sequence in CAPP using an ant colony algorithm. International Journal of Advanced Manufacturing Technology 29 (2006) 1-2, p. 159–164.

KRISHNAMURTI & STOUFFS 2004

Krishnamurti, R.; Stouffs, R.: The boundary of a shape and its classification. Journal of Design Research 4 (2004) 1,

KUSIAK 1990

Kusiak, A.: Intelligent manufacturing systems. Englewood Cliffs, NJ, USA: Prentice-Hall 1990. ISBN: 0-13-468364-1. (Industrial and Systems Engineering).

LAM & DELOSME 1988

Lam, J.; Delosme, J. M.: An efficient simulated annealing schedule: derivation. Report 8816, Department of Computer Science, Yale University (1988).

LEVY et al. 2003

Levy, G. N.; Schindel, R.; Kruth, J. P.: Rapid Manufacturing and Rapid Tooling with Layer Manufacturing (LM) Technologies, State of the Art and Future Perspectives. Annals of the CIRP 52/2/2003 (2003)

LI & MCMAHON 2007

Li, W. D.; McMahon, C. A.: A simulated annealing-based optimization approach for integrated process planning and scheduling. International Journal of Computer Integrated Manufacturing 1 (2007) p. 80-95.

LINDEMANN et al. 2006

Lindemann, U.; Reichwald, R.; Zäh, M. F. (Eds.): Individualisierte Produkte - Komplexität beherrschen in Entwicklung und Produktion. Berlin: Springer 2006. ISBN: 978-3-540-25506-2

LIU et al. 2006

Liu, R.; Zhang, C.; Newman, S. T.: A framework and data processing for interfacing CNC with AP238. International Journal of Computer Integrated Manufacturing 19 (2006) 6, p. 516-522.

MACHINABILITY DATA CENTER 1986

Machinability Data Center: Machining Data Handbook. 3rd ed. Cincinatti, Ohio: Metcut Research Associates 1986. ISBN: 0-936974-02-8. (5th print)

MCCORMACK & CAGAN 2002

McCormack, J. P.; Cagan, J.: Designing inner hood panels through a shape grammar based framework. Artificial Intelligence for Engineering Design, Analysis, and Manufacturing 14 (2002) 4, p. 273-290.

MCCORMACK & CAGAN 2006

McCormack, J. P.; Cagan, J.: Curve-based shape matching: supporting designers' hierarchies through parametric shape recognition of arbitrary geometry. Environment and Planning B: Planning and Design 33 (2006) 4, p. 523-540.

MCMAHON & BROWNE 1993

McMahon, C.; Browne, J.: CADCAM: From principles to practice. Wokingham: Addison-Wesley 1993. ISBN: 0-201-56502-1.

MIAO et al. 2002

Miao, H. K.; Sridharan, N.; Shah, J. J.: CAD-CAM integration using machining features. International Journal for Computer Integrated Manufacturing 15 (2002) 4, p. 296–318.

MICHALEWICZ & FOGEL 2004

Michalewicz, Z.; Fogel, D. B.: How to Solve It: Modern Heuristics. 2nd ed. Berlin: Springer 2004. ISBN: 3-540-22494-7.

MOUNAYRI et al. 1998

Mounayri, H. E.; Spence, A. D.; Elbestawi, M. A.: Milling Process Simulation - A Generic Solid Modeller Based Paradigm. Journal of Manufacturing Science and Engineering 120 (1998) 2, p. 213-222.

NEWELL 1994

Newell, A.: Unified Theories of Cognition. Reprint ed. Cambridge, Massachusetts: Harvard University Press 1994. ISBN: 0-674-92101-1.

NEWMAN et al. 2007

Newman, S. T.; Ali, L.; Brail, A.; Brecher, C.; Klemm, P.; Liu, R.; Nassehi, A.; Nguyen, V. K.; Proctor, F.; Rosso, R. S. U., Jr.; Stroud, I.; Suh, S.-H.; Vitr, M.; Wang, L.; Xu, X. W.: The Evolution of CNC Technology from Automated Manufacture to Global Interoperable Manufacturing. In: ElMaraghy, H. A. et al. (Eds.): Proceedings of the 2nd Int. Conf. on Changeable, Agile, Reconfigurable and Virtual Production (CARV 2007), Toronto, Ontario, Canada, 22-24 July 2007. ISBN: 978-0-9783187-0-3.

ONUH & YUSUF 1999

Onuh, S. O.; Yusuf, Y. Y.: Rapid prototyping technology: applications and benefits for rapid product development. Journal of Intelligent Manufacturing 10 (1999) p. 301-311.

OPEN CASCADE S.A.S. 2011

Open Cascade S.A.S.: Open Cascade Technology, 3d Modeling & Numerical Simulation <http://www.opencascade.org/> 2011-06-05.

OSTROSI & FERNEY 2005

Ostrosi, E.; Ferney, M.: Feature modeling using a grammar representation approach. Artificial intelligence for engineering design, analysis and manufacturing AI EDAM 19 (2005) 4, p. 245-259.

PEARL 1984

Pearl, J.: Heuristics: Intelligent Search Strategies for Computer Problem Solving. Reading, Massachusetts: Addison-Wesley 1984. ISBN: 0-201-05594-5. (Artificial Intelligence). (Reprinted with corrections October, 1985)

PEETERS 2011

Peeters, K.: tree.hh: an STL-like C++ tree class <http://tree.phi-sci.com/> 2011-06-05.

PHAM & DIMOV 2001

Pham, D. T.; Dimov, S. S.: Rapid manufacturing: the technologies and applications of rapid prototyping and rapid tooling. London: Springer 2001.

PIAZZALUNGA & FITZHORN 1998

Piazzalunga, U.; Fitzhorn, P.: Note on a three-dimensional shape grammar interpreter. Environment and Planning B: Planning and Design 25 (1998) 1, p. 11-30.

PREISS & KAPLANSKY 1985

Preiss, K.; Kaplansky, E.: Automated part programming for CNC milling by artificial intelligence techniques. Journal of Manufacturing Systems 4 (1985) 1, p. 51-63.

PUGLIESE & CAGAN 2002

Pugliese, M.; Cagan, J.: Capturing a rebel: modeling the Harley-Davidson brand through a motorcycle shape grammar. Research in Engineering Design 13 (2002) 3, p. 139-156.

PUTZER & ONKEN 2003

Putzer, H.; Onken, R.: COSA – A generic cognitive system architecture based on a cognitive model of human behavior. Cognition, Technology & Work 5 (2003) 2, p. 140-151.

RAHMANI & AREZOO 2007

Rahmani, K.; Arezoo, B.: A hybrid hint-based and graph-based framework for recognition of interacting milling features. Computers in Industry 58 (2007) 4, p. 304-312. (doi: DOI: 10.1016/j.compind.2006.07.001)

REEVES 1995

Reeves, C. R. (Ed.) Modern Heuristic Techniques For Combinatorial Problems. London: McGraw-Hill 1995. ISBN: 0-07-709239-2. (Advanced Topics in Computer Science).

SAKAL & CHOW 1994

Sakal, R. L.; Chow, J. G.: An Integrated Intelligent Process Planning System for Prismatic Parts Using PC-based CAD and CAM Software Packages. International Journal of Advanced Manufacturing Technology 9 (1994) p. 166-174.

SALOMONS et al. 1993

Salomons, O. W.; van Houten, F. J. A. M.; Kals, H. J. J.: Review of research in feature-based design. Journal of Manufacturing Systems 12 (1993) 2, p. 113-132.

SASS 2008

Sass, L.: A physical design grammar: A production system for layered manufacturing machines. Automation in Construction 17 (2008) 6, p. 691-704. (doi: DOI: 10.1016/j.autcon.2007.12.003)

SCHUH et al. 2007

Schuh, G.; Klocke, F.; Brecher, C.; Schmitt, R. (Eds.): Excellence in Production - Festschrift für Univ.-Prof. Dr.-Ing. Dipl.-Wirt. Ing. Dr. techno h.c. Dr. oec. h.c. Walter Eversheim. Aachen: Apprimus 2007. ISBN: 978-3-940565-00-6.

SHAH 1988

Shah, J. J.: Feature transformations between application-specific feature spaces. Computer-Aided Engineering Journal 5 (1988) 6, p. 247-255.

SHAH et al. 2001

Shah, J. J.; Anderson, D.; Kim, Y. S.; Joshi, S.: A Discourse on Geometric Feature Recognition From CAD Models. Journal of Computing and Information Science in Engineering 1 (2001) 1, p. 41-51.

SHAH et al. 1995

Shah, J. J.; Balakrishnan, G.; Rogers, M. T.; Urban, S. D.: Comparative Study of Procedural and Declarative Feature Based Geometric Modeling. In: Soenen, R. et al. (Eds.): Advanced CAD/CAM Systems - State-of-theart and future trends in feature technology. London: Chapman-Hall 1995, ISBN: 0-412-61730-7.

SHAH & MÄNTYLÄ 1995

Shah, J. J.; Mäntylä, M.: Parametric and feature-based CAD/CAM: concepts, techniques, and applications. New York: John Wiley & Sons 1995. ISBN: 0-471-00214-3.

SHAH et al. 1994a

Shah, J. J.; Mantyla, M.; Nau, D. S. (Eds.): Advances in Feature Based Manufacturing. Amsterdam, The Netherlands: Elsevier 1994a. ISBN: 0-444-81600-3. (Manufacturing Research and Technology 20).

SHAH et al. 1994b

Shah, J. J.; Mäntylä, M.; Nau, D. S.: Introduction To Feature Based Manufacturing. In: Shah, J. J. et al. (Eds.): Advances in Feature Based Manufacturing. Amsterdam, The Netherlands: Elsevier 1994b, p. 1-11. ISBN: 0-444-81600-3. (Manufacturing research and technology 20).

SHEA 1997

Shea, K.: Essays of Discrete Structures: Purposeful Design of Grammatical Structures by Directed Stochastic Search. Ph.D. dissertation, Carnegie-Mellon University, Pittsburgh (1997).

SHEA & CAGAN 1999

Shea, K.; Cagan, J.: Languages and semantics of grammatical discrete structures. Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AIEDAM) 13 (1999) 4, p. 241-251.

SHEA et al. 2008

Shea, K.; Engelhard, M.; Ertelt, C.; Hoisl, F.: A Framework for a Cognitive Design-to-Fabrication System, Proceedings of the 7th International Symposium on Tools and Methods of Competitive Engineering (TMCE 2008). Ismir, Turkey, 21. – 25. April 2008 2008.

SHEA et al. 2010

Shea, K.; Ertelt, C.; Gmeiner, T.; Ameri, F.: Design-to-fabrication automation for the cognitive machine shop. Advanced Engineering Informatics 24 (2010) 3, p. 251-268.

SHIRASE & FUJII 2009

Shirase, K.; Fujii, S.: Machine Tool Automation. In: Nof, S. Y. (Ed.) Springer Handbook of Automation. Berlin: Springer 2009, ISBN: 9783540788300.

SHIRUR et al. 1998

Shirur, A.; Shah, J. J.; Hirode, K.: Machining algebra for mapping volumes to machining operations for developing extensible generative CAPP. Journal of Manufacturing Systems 17 (1998) 3, p. 167-182.

SIMON & NEWELL 1958

Simon, H. A.; Newell, A.: Heuristic Problem Solving: The Next Advance in Operations Research. Operations Research 6 (1958) 1, p. 1-10.

SMITH & WRIGHT 1996

Smith, C. S.; Wright, P. K.: CyberCut: A World Wide Web based design-to-fabrication tool. Journal of Manufacturing Systems 15 (1996) 6, p. 432-442.

SPENCE & ALTINTAS 1994

Spence, A. D.; Altintas, Y.: A Solid Modeller Based Milling Process Simulation and Planning System. Journal of Engineering for Industry 116 (1994) 1, p. 61-69.

STARLING & SHEA 2005

Starling, A. C.; Shea, K.: A parallel grammar for simulation-driven mechanical design synthesis. In: Proceedings of the ASME International Design Engineering Technical Conferences & Computers and Information in Engineering Conference IDETC/CIE, Long Beach, California, USA, September 24-28 2005.

STINY 1980

Stiny, G.: Introduction to shape and shape grammars. Environment and Planning B: Planning and Design 7 (1980) 3, p. 343-351.

STINY 1982

Stiny, G.: Spatial relations and grammars. Environment and Planning B: Planning and Design 9 (1982) 1, p. 113-114.

STINY 1991

Stiny, G.: The algebras of design. Research in Engineering Design 2 (1991) 3, p. 171-181.

STINY 2006

Stiny, G.: Shape: talking about seeing and doing. Cambridge, MA, USA: MIT press 2006. ISBN: 0-262-19531-3.

STINY & GIPS 1972

Stiny, G.; Gips, J.: Shape grammars and the generative specification of painting and sculpture. In: Freiman, C. V. et al. (Eds.): Proceedings of the IFIP Congress 71 (Information Processing 71), Ljubljana, Yugoslavia, Aug. 23 - 28, 1971 1972. North-Holland Publishing Company p. 1460-1465. ISBN: 0-7204-2063-6.

STOUFFS & KRISHNAMURTI 2006

Stouffs, R.; Krishnamurti, R.: Algorithms for classifying and constructing the boundary of a shape. Journal of Design Research 5 (2006) 1, p. 54-95.

SUNDARARAJAN & WRIGHT 2007

Sundararajan, V.; Wright, P.: CyberCut: A Coordinated Pipeline of Design, Process Planning and Manufacture. In: Wang, L. et al. (Eds.): Process Planning and Scheduling for Distributed Manufacturing. London: Springer 2007, ISBN: 978-1-84628-751-0. (Springer Series in Advanced Manufacturing).

VANDENBRANDE & REQUICHA 1993

Vandenbrande, J. H.; Requicha, A. A. G.: Spatial Reasoning for the Automatic Recognition of Machinable Features in Solid Models. IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (1993) 12, p. 1269-1285.

VERNON et al. 2007

Vernon, D.; Metta, G.; Sandini, G.: A Survey of Artificial Cognitive Systems: Implications for the Autonomous Development of Mental Capabilities in Computational Agents. IEEE Transactions on Evolutionary Computation 11 (2007) 2, p. 151-180.

WANG & DUARTE 2002

Wang, Y.; Duarte, J. P.: Automatic generation and fabrication of designs. Automation in Construction 11 (2002) 3, p. 291-302.

WECK & WOLF 2003

Weck, M.; Wolf, J.: Introduction to STEP-NC, A standard providing data for modern NC-machining enabling enhanced functionality. In: STEP NC-Seminar. Aachen, Germany: 2003.

WRIGHT et al. 2004

Wright, P. K.; Dornfeld, D. A.; Sundararajan, V.; Misra, D.: Tool Path Planning Generation For Finish Machining of Freeform Surfaces in the Cybercut Process Planning Pipeline, NAMRC XXXII. 2008-07-09 2004. <http://repositories.cdlib.org/lma/codef/wright_04_1> -

XU & HE 2004

Xu, X.; He, Q.: Striving for a total integration of CAD, CAPP, CAM and CNC. Robotics and Computer Integrated Manufacturing 20 (2004) 2, p. 101-110.

YAO et al. 2007

Yao, S.; Han, X.; Yang, Y.; Rong, Y. K.; Huang, S. H.; Yen, D. W.; Zhang, G.: Computer aided manufacturing planning for mass customization: part 2, automated setup planning. International Journal of Advanced Manufacturing Technology 32 (2007) 1, p. 205-217.

YIN & CAGAN 2000

Yin, S.; Cagan, J.: An Extended Pattern Search Algorithm for Three-Dimensional Component Layout. Journal of Mechanical Design 122 (2000) 1, p. 102-108.

ZEID 1991

Zeid, I.: CAD/CAM Theory and Practice. New York: McGraw-Hill 1991. ISBN: 0-07-072857-7. (McGraw-Hill Series in Mechanical Engineering).

ZHANG et al. 2006

Zhang, C.; Liu, R.; Hu, T.: On the futuristic machine control in a STEP-compliant manufacturing scenario. International Journal of Computer Integrated Manufacturing 19 (2006) 6, p. 508-515.

ZHANG & ALTING 1994

Zhang, H.-C.; Alting, L.: Computerized Manufacturing Process Planning Systems. 1st ed. London: Chapman & Hall 1994. ISBN: 0-412-41300-0.

9 Appendix

9.1 Software usage

After starting the application, the main window is displayed. Any other window can be closed. The main menu is at the top of the window. Here, the SGMP menu is most important. Below the controls can be used to create and remove geometries as well as change the visualization style and manipulate the viewpoint. The black area is the display for visualization.

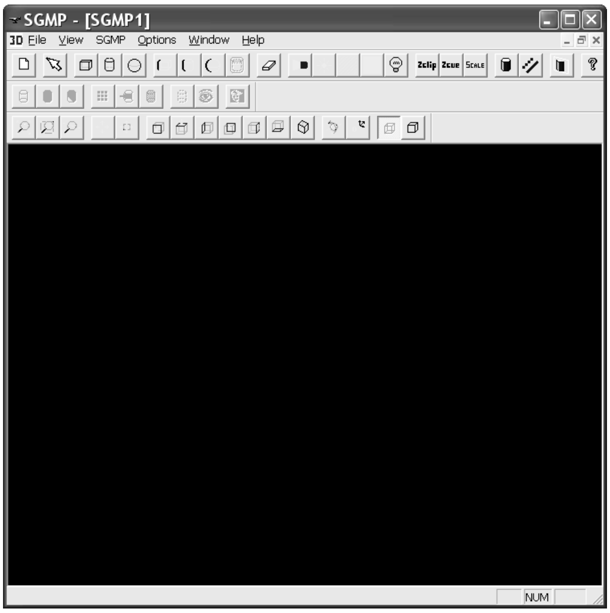
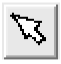
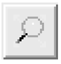
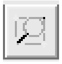
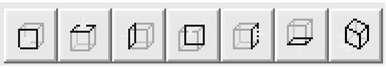


Figure 9-1: Main window based on the Viewer3d application.

9.1.1 Main window controls

The most important control elements within the main window are described in Table 9-1.

Table 9-1: Main window controls.

	Toggle selection mode on/off. After application start selection mode is off. Running SGMP with selection mode off is recommended, otherwise application can crash.
	Fit view to window. Adjusts zoom to display all objects at once.
	Zoom to selection. Draw a selection box on the view to magnify.
	Zoom using the mouse.
	Switch to pre-defined view-points (from left to right): front, top, left, back, right, bottom or isometric view.

All other controls to change the visualization style are functional. However it is not recommended to use controls that create or remove objects other than those from SGMP.

All functions to setup and start jobs for SGMP are in the SGMP menu shown in Figure 9-2. It is possible to load from a STEP file a part design and a raw workpiece, select a workpiece from the ontology, and generate the TRV from part and raw shape. Solid volume to be removed is displayed and the boundary between the TRV and the part design is displayed in green. Machining plans can be generated using different mechanisms: Generate Plan using Simulated Annealing Outer Loop, using A* and using A* with Simulated Annealing re-optimization. Finally, the resulting TRV can be saved as STEP file for further inspection. Usually using A* only delivers good results in a reasonable time.

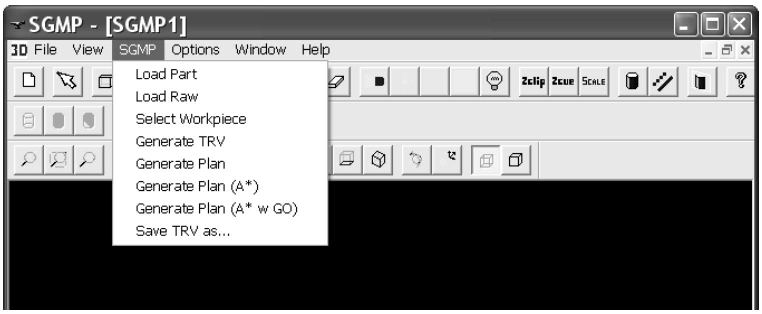


Figure 9-2: SGMP menu.

9.1.2 Running machining planning processes

Creating and running a machining planning should be done using the following procedure:

1. Load a part design using the SGMP menu.
2. Define the workpiece shape by either loading the workpiece shape or selecting a workpiece from an ontology
3. Generate and visually inspect the TRV.
4. Generate the machining plan (it is recommended to use A*)
5. Visually inspect the result and save the TRV.

For better inspection the selection mode can be set to on to highlight edges of the TRV. The NC-code is written to the "D:\" location as text file if the appropriate pre-compile command is defined. However, before transferring it to a machine tool it must be checked for correctness.

9.1.3 File structure of the development environment:

D:
MSDL-Fullv1.owl → MSDL Ontology for workpiece selection
WorkpieceMatchmaker.jar → Workpiece matchmaker, interface to query the ontology
D:\devel
OpenCASCADE6.3.0 → OpenCASCADE directory
SVN
CCalc → source directory for command line SGMP calculation application
Common → common includes of the OpenCASCADE demo samples
loki-0.1.7 → source directory of the loki library
MP1 → source directory of fully implemented SGMP for 3-axis milling
SGMP → source directory of abstract SGMP implementation
Tree → source directory of tree structure
View_Anim_v1 → source directory of the GUI for SGMP based on Viewer3d

9.1.4 Pre-compiler directives

The pre-compiler directives widely control the behavior of SGMP. The directives can be defined or undefined in the **SGMP.h** source file. The directives are listed below

Table 9-2: Pre-compiler directives.

SAOUTERITERATIONS (default 100)	Defines the number of Simulated Annealing (SA) iterations in the outer loop (rule selection).
SAOUTERMOVES (default 300)	Defines the number of Simulated Annealing (SA) moves in the outer loop (rule selection).
SAREOPTITERATIONS (default 100)	Defines the number of Simulated Annealing (SA) iterations during re-optimization.
SAREOPTMOVES (default 100)	Defines the number of Simulated Annealing (SA) moves during re-optimization.
ASTAREXPANDEDSTATES (default 500)	Defined the number of states that are explored during A* search for rule selection before terminating.
MAXCUTTINGDEPTHMULT (default 0.5)	Defines the factor to calculate the maximum cutting depth from the tool-diameter.
PSMAXITERATIONS (default 100)	Defines the number of Patters Search iterations in the inner loop, i.e. during parameter optimization.
GOSTATS (default undefined)	Define this directive to enable detailed statistics for re-optimization
CONSTW (default 0.5)	Defines the weight to balance between h(x) and g (x) in f(x) = w*g(x) + (1-w)*h(x).
CONSTWP (default 1.0)	Defines the weight assigned to the penalty p(x).
CONSTPOFFSET (default 10.0)	Defines the offset (base value) to calculate the penalty p(x).
CONSTPEXP (default 2)	Defines the exponent to calculate the penalty (p(x))^exp.

9.1.5 Activating and deactivating grammar elements

The elements of the grammar can be activated or deactivated by registering or unregistering them in the **SGMPMP1.h** source file.

9.2 Spatial grammar machining planning (SGMP) framework documentation

The following documentation of the Spatial Grammar Machining Planning (SGMP) system is based on the auto-generated doxygen documentation. The full doxygen documentation can be created from the SGMP source code.

9.2.1 Modules

The SGMP system consists of two modules: The abstract core functionality and a derived implementation of SGMP for 2.5D milling.

Spatial Grammar Machining Planning (SGMP)

This module implements the core functionality of SGMP. However, due to its abstract nature, this module cannot be compiled directly. To use it, derived classes must be created, that implement valid functionality. Figure 9-3 shows a UML diagram of the SGMP core module displaying the relation of the classes and selected core member functions.

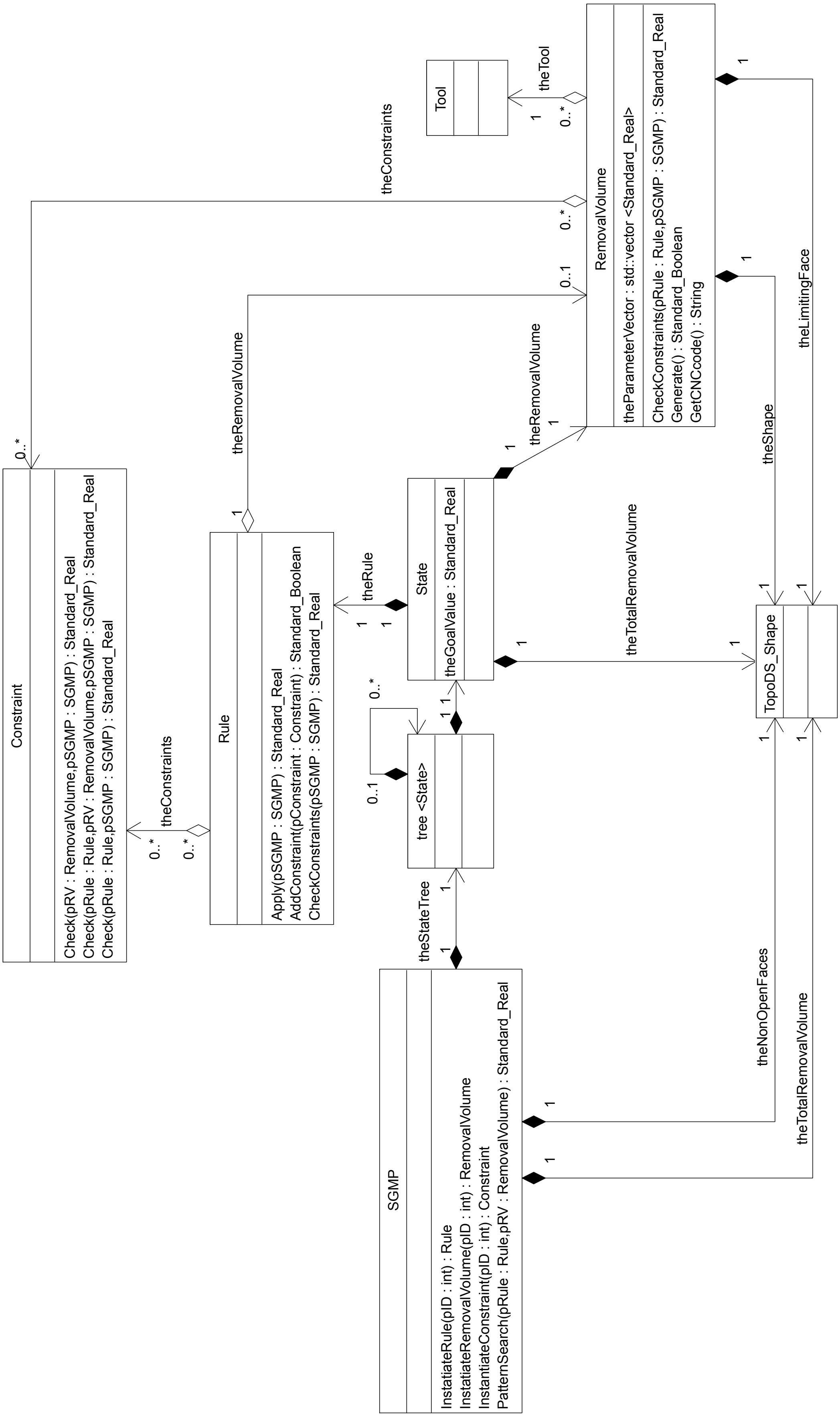


Figure 9-3: UML diagram of the SGMP module reduced to core functionality

SGMP for 2.5D milling

This module implements the crucial functionality of the SGMP focused on 2.5D milling. All rules and vocabulary is defined in this module.

9.2.2 Class hierarchy

- Constraint
  - CDifferentTool
  - CIsMarkerSet
  - CLFTRVnonintersec
  - CLimitComputationTime
  - CNOFLFnoncoin
  - CNOFRVnoninter
  - CNoRepRules
  - CRVPartNonintersect
  - CSameTool
  - CWithinDistance
- MachineTool
- QualityItem
- QualityList
- RemovalVolume
  - RVcslotD10
  - RVcylD10
    - RVcylD15
  - RVslotD10
    - RVslotD15
- Rule
  - RApplyRV
  - RChangeTool
  - RResetMarker
  - RStartingSymbol
  - RTerminal
  - RuleSequence
    - RSChangeTooltoD10
    - RSChangeTooltoD15
- SGMP
  - SGMPMP1
- State
- Tool
- Workpiece

9.2.3 Class list

In the following the classes of the framework are briefly described.

CDifferentTool

Constraint to use different tools. This Constraint checks whether the tool that is used in the RemovalVolume is different from the currently used tool (in the workspace) Returns >= 0.0 if they are different.

CIsMarkerSet

Constraint to check whether a marker has been set .This Constraint checks whether a marker has already been set within the workspace environment Returns >= 0 if marker has been set.

CLFTRVnonintersec

Constraint avoiding intersection of Limiting Face (LF) and Total Removal Volume (TRV). This Constraint checks whether the Limiting Face (LF) of the Removal Volume intersects with the Total Removal Volume (TRV). This prevents exceeding the maximum cutting depth. Returns >= 0 if not intersecting.

CLimitComputationTime

Constraint aimed at limiting the computation time during applying RemovalVolumes.

CNOFLFnoncoin

Constraint avoiding intersection between Non-Open Faces and Limiting Face (LF). This Constraint checks whether the Non-Open Faces (NOF) and the Limiting Face (LF) of the Removal Volume are intersecting. Returns >= 0 if they are not intersecting.

CNOFRVnoninter

Constraint to avoid intersection between Non-Open Faces and Removal Volume. This Constraint checks whether the Non-Open Faces (NOF) and Removal Volume (RV) are intersecting. Returns >= 0 if they are not intersecting.

CNoRepRules

Constraint to use rules only once in a row. This Constraint checks whether the rule is applied twice in a row. This can be used to prevent the ineffective application of rules. Returns >= 0 if the rules are not repetitive.

Constraint

The class Constraint implements constraints that can be used during search.

CRVPartNonintersect

Constraint avoiding intersection between Removal Volume and part. This Constraint checks whether the Removal Volume (RV) and the later part shape are intersecting. Returns >= 0 if they are not intersecting.

CSameTool

Constraint for using the same tool. This Constraint checks whether the tool that is used in the RemovalVolume is the same as the currently used tool (in the workspace) Returns >= 0.0 if they are identical.

CWithinDistance

Constraint to prevent growth of Removal Volumes This Constraint checks whether the Removal Volume (RV) fits inside the bounding box of the Total Removal Volume (TRV). This can be used to prevent Removal Volumes of becoming too big. Returns >= 0.0 if the Removal Volume fits inside the bounding box of the TRV.

MachineTool

Implements a model of a machine tool. This class represents a basic model for the machine tools, including which tools can be hold at which position.

QualityItem

Aggregates data for Hustin moves during Simulated Annealing. This class collects all information required to calculate Hustin moves (i.e. rule probabilities) for Simulated Annealing.

QualityList

Manages data for Hustin moves during Simulated Annealing. This class acts as container for QualityItems to calculate Hustin moves (i.e. rule probabilities) for Simulated Annealing.

RApplyRV

Rule to apply Removal Volumes to the TRV. This rule applies a Removal Volume to the shape of the Total Removal Volume. The call to the GetCNCcode method is redirected to the attached RemovalVolume.

RChangeTool

Rule to perform a tool change. This rule performs a tool change and instantiates the required CNC code.

RemovalVolume

Implements the volume removed during a machining operation. This class represents the volume that can be removed from a workpiece by a single machining operation.

RResetMarker

Rule to reset the marker, i.e. reposition the tool. This rule resets the marker. This corresponds to retracting the tool, transferring and to a different location at rapid feed. The new location is determined by choosing a random position above the bounding box of the Total Removal Volume (TRV).

RSChangeTooltoD10

Rule Sequence to actually perform a tool change. This rule sequence performs a tool change to the D10 tool by applying the RChangeTool rule, and then applying the RApplyRV rule. Through this, the search accepts the RChangeTool rule that primarily only causes cost.

RSChangeTooltoD15

Rule Sequence to actually perform a tool change. This rule sequence performs a tool change to the D15 tool by applying the RChangeTool rule, and then applying the RApplyRV rule. Through this, the search accepts the RChangeTool rule that primarily only causes cost.

RStartingSymbol

Rule to apply the Starting Symbol (initial rule). This rule is the starting symbol. It initializes the workspace and defines the orientation of the envelope. The marker is placed above the centroid of the bounding box surrounding the Total Removal Volume (TRV).

RTerminal

Rule to apply the terminal. This rule ends the planning process. The marker is removed from the workspace and the tool retracted to a save position.

Rule

This class represents rules within the grammar framework.

RuleSequence

This class represents a rule sequence within the grammar framework. With this class several rules can be applied at once.

RVcslotD10

Removal Volume for a curved slot with D=10mm. This RV represents a Removal Volume created by a curved toolpath and a tool with D=10mm. (experimental)

RVcylD10

Removal Volume for a cylinder with D=10mm. This RV represents a Removal Volume created by a line toolpath and a tool with D=10mm where the toolpath and tool axis are co-linear

RVcylD15

Removal Volume for a cylinder with D=15mm. This RV represents a Removal Volume created by a line toolpath and a tool with D=15mm where the toolpath and tool axis are co-linear.

RVslotD10

Removal Volume for a slot with D=10mm. This RV represents a Removal Volume created by a line toolpath and a tool with D=10mm where the toolpath and tool axis are perpendicular.

RVslotD15

Removal Volume for a slot with D=15mm. This RV represents a Removal Volume created by a line toolpath and a tool with D=15mm where the toolpath and tool axis are perpendicular.

SGMP

The class SGMP (short for Spatial Grammar Machining Planning) represents the abstract main class.

SGMPMP1

The class SGMPMP 1 (Machining Planning 1) represents the core of the implemented SGMP for 2.5D milling.

State

Represents a state during search. This class represents a state of the workspace, i.e. the current configuration and the applied rule and Removal Volume.

Tool

This class represents a tool used for cutting processes.

Workpiece

Implements a workpiece model. This class represents a workpiece/raw material used as input to the planning system.