

Entwicklung einer Werkzeugverwaltung für die Frässimulation 'CNC Milling'

Carsten Witte

Thema: Entwicklung einer Werkzeugverwaltung für Fräswerkzeuge für die Simulation
'CNC Milling'

Verfasser: Carsten Witte
Matrikelnummer: 150823
Fachbereich Automatisierungstechnik

Hochschule: Fachhochschule Nordostniedersachsen
Volgershall 1
21339 Lüneburg

Firma: STA Softwareentwicklung für Technische Anwendungen GmbH
Rudower Chaussee 29
12489 Berlin

Abgabedatum: 18. August 2003

Betreuer: 1. Gutachter: Prof. Dr.-Ing. Ralph Welge
2. Gutachter: Prof. Dr.-Ing. Dipl.-Inform. Eckhard C. Bollow
Projektbetreuer bei STA: Dipl.-Math. techn. Axel Friedrich

Ziel der Arbeit: Ziel dieses Projekts ist der Entwurf eines Klassifizierungssystems für Fräswerkzeuge, sowie die Entwicklung einer darauf aufbauenden Werkzeugverwaltung. Diese soll die Möglichkeit bieten, automatisch Werkzeugmodelle aus den gespeicherten Daten zu generieren. Das Darstellungsformat der Werkzeugmodelle muss dabei direkt von der Simulation 'CNC Milling' lesbar sein.

Ergebnisse: Die, im Rahmen der Diplomarbeit, erstellte Software soll von der STA GmbH als Zusatzmodul zur Simulation 'CNC Milling' angeboten werden

Inhalt

1	Einleitung.....	5
1.1	Firmenvorstellung.....	5
1.2	Ziel dieses Projekts.....	6
1.3	Über dieses Dokument.....	6
2	Stand der Technik.....	7
2.1	Ablauf einer Frässimulation.....	7
2.2	Verwendete Dateiformate.....	9
2.3	Marktvergleich von Werkzeugverwaltungen.....	13
3	Problembeschreibung und Lösung.....	15
3.1	Entwicklung eines Klassifizierungssystems.....	15
3.1.1	Entwerfen einer Methodik zur Klassifizierung.....	16
3.1.2	Richtlinien zur Auswahl der Parameter.....	17
3.1.3	Erläuterung der Parameter.....	19
3.2	Spezifikation der Software.....	24
3.2.1	Anwendungsfalldiagramm.....	24
3.2.2	Generieren von GD-Dateien aus Datensätzen.....	32
4	Implementierung.....	33
4.1	Auswahl der Hilfsmittel.....	33
4.2	Klassendiagramme.....	35
4.2.1	Datenmodell.....	35
4.2.2	Klassendiagramm der Hauptanwendung.....	38
4.3	Erläuterungen zum Quellcode.....	39
4.3.1	Dialogstruktur.....	39
4.3.2	Die Klasse CODBCWrapper.....	45
4.4	TCL-Skripte.....	48
4.4.1	Verwendung der Tcl-Skripte.....	48
4.4.2	Erläuterung der einzelnen Skripte.....	48
4.5	Zeichnungen zu Werkzeugen und Aufnahmen.....	53
5	Zusammenfassung und Ausblick.....	54
A)	Glossar.....	55
B)	Abkürzungsverzeichnis.....	56
C)	Abbildungsverzeichnis.....	57
D)	Quellen.....	58
E)	Beispiel zur Verwendung der Werkzeugverwaltung.....	60

Danksagung

An dieser Stelle möchte ich mich sehr herzlich bei Dr. Carsten Erdmann bedanken, der mir bei der Auswahl des Diplomthemas sehr entgegengekommen ist, sowie bei Dipl.-Math. Axel Friedrich und allen anderen Mitarbeitern der STA, die mir jederzeit mit kompetentem Rat weiterhelfen konnten.

Des weiteren bedanke ich mich bei Prof. Dr. Welge und Prof. Dr. Bollow, die dieses Diplom betreuen und mir mit vielen guten Ratschlägen sehr bei der Durchführung halfen.

Mein besonderer Dank gilt Prof. Dr. Ulrich Berger von der BTU Cottbus, der für mich den Kontakt zur STA GmbH geknüpft hat.

1 Einleitung

1.1 Firmenvorstellung

Die STA Softwareentwicklung für Technische Anwendungen GmbH wurde 1996 von Dr. Carsten Erdmann und Dr. Bernd Rämmer in Berlin gegründet. Heute beschäftigt sie insgesamt zwölf Mitarbeiter und entwickelt hauptsächlich Simulationen für Dreh- und Fräsbearbeitungen ('CNC Turning' und 'CNC Milling'). Die Software 'CNC Turning' ist bereits auf dem Markt etabliert und wird von namenhaften Firmen wie der Gildemeister AG vertrieben.

Der Vorteil solcher Simulationen liegt darin, dass eine Werkstückbearbeitung 'offline', also ohne Verwendung einer realen Maschine und in kurzer Zeit durchgeführt werden kann. So können eventuelle Bearbeitungsfehler, wie z.B. Kollisionen in der Maschine, frühzeitig erkannt und behoben werden. Des Weiteren ermöglicht die Verwendung der Simulation Kosteneinsparungen durch eine mögliche Reduzierung von Rüst- und Bearbeitungszeiten.

Eine marktreife Basisversion der Frässimulation 'CNC Milling' ist weitgehend fertiggestellt. Es handelt sich dabei um eine Simulation für komplexe Mehrachs-Fräsoptionen, die eine 3D-Visualisierung von Zerspanprozessen, inklusive Kollisionsprüfung und Berechnung der Materialentnahme bietet. Die Simulation basiert auf 3D-Modellen, welche sowohl die Kinematik, als auch die Geometrie von realen Bearbeitungszentren weitgehend originalgetreu wiedergeben. Abbildung 1 zeigt das Modell einer simulierten Fräsmaschine.



**ABBILDUNG 1. Bearbeitungszentrum DMU 80P der Firma Deckel Maho
Quelle: STA GmbH**

1.2 Ziel dieses Projekts

Für jede Fräsbearbeitung eines Werkstücks ist ein Satz an Werkzeugmodellen nötig, der für die Simulation bereitgestellt werden muss. Da die Zahl der existierenden Fräswerkzeuge nahezu unüberschaubar groß ist, macht es Sinn, die Werkzeuge zu klassifizieren und in einer geordneten Struktur zu verwalten.

Bisher existiert weder eine solche Klassifizierung, noch ein Mechanismus zur Verwaltung von Werkzeugmodellen. Des Weiteren muss jedes Werkzeug manuell erstellt werden.

Ziel dieses Projekts ist deshalb zum einen der Entwurf einer Klassifizierung für alle Arten von Fräswerkzeugen, zum anderen die Entwicklung einer Software, die diese Werkzeuge verwalten kann. Die manuelle Erstellung der Werkzeugmodelle soll durch einen, von der Werkzeugverwaltung bereitgestellten Automatismus, ersetzt werden.

1.3 Über dieses Dokument

Kapitel 2 erläutert die Konfiguration eines Simulationsprojekts und alle dafür nötigen Dateien, so dass ein Überblick über den aktuellen Stand der Simulation geschaffen wird. Es folgt eine Beschreibung der Anforderungen an die Werkzeugverwaltung, mit anschließender Spezifikation des Systems. Kapitel 4 liefert eine Beschreibung der Implementation.

Im Anhang finden sich Quellenangaben, sowie ein Glossar für Fachbegriffe und Abkürzungen, die nicht im Haupttext erklärt werden. Am Ende dieses Diploms ist ein bebildertes Anwendungsbeispiel zur Werkzeugverwaltung angefügt, um einen Eindruck vom fertigen Programm zu vermitteln.

2 Stand der Technik

Dieses Kapitel gibt einen Überblick über den jetzigen Stand der Software 'CNC Milling' und beschreibt die Schritte, die für die Simulation eines Fräsprojekts durchgeführt werden.

2.1 Ablauf einer Fräsimulation

Zur Durchführung einer Simulation gehören eine Reihe von Komponenten, die in Abbildung 2 dargestellt sind.

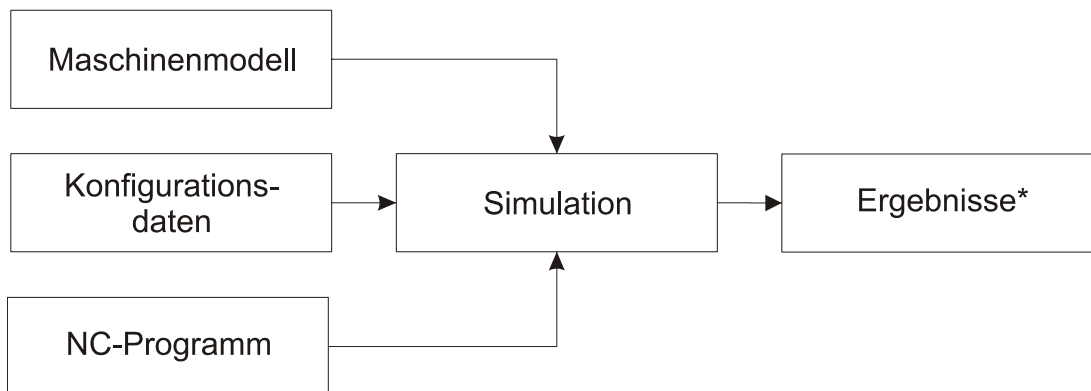


ABBILDUNG 2. Komponenten von 'CNC Milling'

* getestetes NC-Programm, Bearbeitungszeiten, Einrichtdaten, u.s.w

Dieses Unterkapitel gibt einen Überblick über den generellen Simulationsablauf, indem die Schritte aufgezählt werden, die für die Durchführung eines Simulationsprojekts nötig sind. Die dabei verwendeten Dateitypen werden in Abschnitt 2.2 genauer beschrieben.

1. NC-Datei bereitstellen.

NC-Dateien enthalten alle Maschinenbefehle zur Bearbeitung eines Werkstücks. Sie werden i.d.R. nicht manuell, sondern durch sog. CAD/CAM-Systeme erstellt. Dies geschieht, indem ein 3D-Modell des Werkstücks mit einem CAD-Programm erstellt wird und danach mit einem CAM-System in NC-Code umgewandelt wird.

2. Maschinen-GD-Datei bereitstellen.

In der Maschinendatei ist ein originalgetreues Modell eines Bearbeitungszentrums abgelegt (siehe Abb.1), welches auch die Kinematik der Maschine berücksichtigt. Diese Modelle werden von der STA GmbH erstellt, wobei technische Zeichnungen und CAD-Modelle der Maschinenhersteller als Vorlagen dienen.

3. Konfigurationsdatei und Werkzeug-GD-Dateien bereitstellen.

Wie in Abschnitt 2.2 beschrieben, werden in der Konfigurationsdatei Werkzeuge bestimmt, die für das Projekt verwendet werden. Die Informationen über die zu verwendenden Werkzeuge können einem sogenannten Einrichtblatt entnommen werden, das i.d.R. jedem Fräsprojekt beiliegt. Sind nicht alle benötigten Werkzeugmodelle vorhanden, so müssen für die Fehlenden neue GD-Dateien erstellt werden. Im Gegensatz zu den Maschinenmodellen, werden die Werkzeuge jedoch nicht von der STA GmbH, sondern von einer Partnerfirma namens BAMiTec AG ([BAMiTec 03]) erstellt.

4. Projektdatei erstellen.

Hier werden die Namen von NC-Datei, Maschinendatei und Konfigurationsdatei eingetragen, die für die Simulation verwendet werden sollen.

5. Durchführen der Simulation.

Sollten im Verlauf der Simulation Kollisionen oder andere Fehler entdeckt werden, kann die NC-Datei entsprechend korrigiert werden. Am Ende des Prozesses steht eine getestete NC-Datei, die direkt für die reale Fräsbearbeitung verwendet werden kann.

Abbildung 3 zeigt den Arbeitsbereich einer vollständig konfigurierten Fräsmaschine. Man sieht am oberen Rand ein Werkzeug in einer Spindel und im Zentrum ein eingespanntes, quaderförmiges Rohteil.

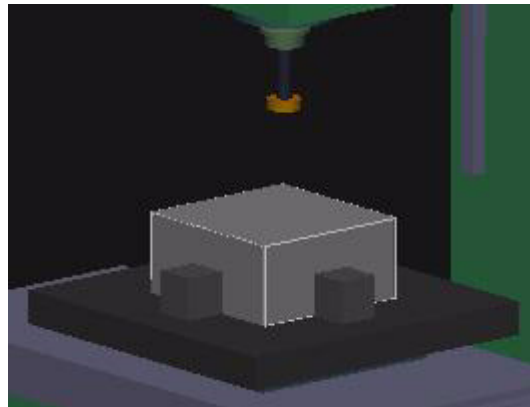


ABBILDUNG 3. konfigurierte Maschine
Quelle: STA GmbH

Wie in Arbeitsschritt 3 beschrieben, müssen alle zu simulierenden Werkzeuge manuell modelliert werden. Dieser Vorgang kann i.d.R. nicht vom Kunden selber durchgeführt werden und muss deshalb von einer externen Firma erledigt werden, was Zeit und Geld kostet.

Ziel dieses Projekts ist es, ein System zu entwickeln, mit dem die Werkzeugmodellierung und -verwaltung vor Ort und ohne Spezialwissen durchführbar ist.

2.2 Verwendete Dateiformate

Geometry Data (GD)

GD-Dateien sind registriert als 'STA Geometry Data files' und können dreidimensionale Objekte jeder Art beschreiben - zum Beispiel Werkstücke, Werkzeuge, Spannmittel aber auch ganze Werkzeugmaschinen, samt ihren kinematischen Eigenschaften.

Sie bestehen aus hierarchisch angeordneten Knoten, von denen jeder von einem bestimmten Typ ist und eine Reihe von Attributen besitzt. Eine genaue Spezifikation ist in [STA 03] zu finden. Ein Werkzeug, das durch eine GD-Datei beschrieben wird, besteht immer aus drei Komponenten: der Schneide (insert), dem Schaft (holder) und der Werkzeugaufnahme (clamp).

Das folgende Beispiel zeigt eine GD-Datei, die einen Schaftfräser in einer Aufnahme (SK 30) beschreibt. Die Werkzeugverwaltung soll genau diese Art von GD-Datei erzeugen, weshalb der Quelltext in voller Länge dargestellt ist.

```
#STA GD V2.0
Tool {
  nosePoint 0 0 -69
  noseRadius 4
  insert Transform {
    children Transform {
      translation 0 0 -90
      rotation 0 -1 0 90
      children Shape {
        material Material {
          color 0.7 0 0
          specularIntensity 0.6
        }
        geometry Revolve {
          contour [0 0 0 0 0 0,
                 0 0 4.0 0 0 0,
                 -1 0 4.0 0 4.0 0,
                 0 19 4.0 0 0 0,
                 0 19 0 0 0 0]
        }
      }
    }
  }
  holder Transform {
    children Transform {
      translation 0 0 -90
      rotation 0 -1 0 90
      children Shape {
```

```

    material Material {
      color 0.15 0.15 0.2
    }
    geometry Revolve {
      contour [0 19 0 0 0 0,
              0 19 4.0 0 0 0,
              0 24 5.0 0 0 0,
              0 69 5.0 0 0 0,
              0 69 0 0 0 0]
    }
  }
}
clamp Transform {
  children Transform {
    rotation 0 -1 0 90
    children Shape {
      material Material {
        color 0.3 0.4 0.3
      }
      geometry Revolve {
        contour [0 -50 0 0 0 0,
                0 -50 12.5 0 0 0,
                0 -39.9 17.5 0 0 0,
                0 -9.6 17.5 0 0 0, 0 -9.6 25 0 0 0,
                0 -1.6 25 0 0 0,
                0 -1.6 15.875 0 0 0,
                0 0 15.875 0 0 0,
                0 48.4 8.7 0 0 0,
                0 68.4 8.7 0 0 0,
                0 68.4 0 0 0 0]
      }
    }
  }
}
}

```

Der Basisknoten ist vom Typ 'Tool', welcher in [STA 99] spezifiziert wird. Die wichtigsten Elemente dieses Knotens sind 'insert', 'holder' und 'clamp', die alle vom Typ 'Transform' sind. Dieser Typ stellt einen Knoten dar, der seinerseits beliebige Knoten (children) aufnehmen kann und die Eigenschaften 'translation' und 'rotation' besitzt. Für jeden der drei Transformknoten ist eine Rotation von 90° um die Y-Achse definiert (`rotation 0 -1 0 90`), damit das Werkzeug in der Frässimulation richtig ausgerichtet ist. Die Berechnung der Verschiebung für Schneide und Halter ist in Kapitel 4.4.2 näher beschrieben. Eine Drehung oder Verschiebung eines Transformknotens wirkt

sich auf alle untergeordneten Objekte aus.

Die eigentliche Werkzeuggeometrie wird durch einen Rotationskörper (geometry Revolve) dargestellt, der aus einer 2D-Kontur (contour) erzeugt wird. Die Syntax zur Beschreibung der einzelnen Wegstrecken einer Kontur ist in [STA 03] definiert.

Numerical Control (NC)

NC-Dateien enthalten die zu simulierenden Fräsprogramme und sind damit ein wesentlicher Bestandteil eines Simulationsprojekts. Sie beinhalten Anweisungen zur Steuerung einer Werkzeugmaschine, zum Beispiel Verfahrbefehle, Änderungen des Koordinatensystems, Werkzeugwechselbefehle, Kommandos an die Spindel oder Ähnliches.

Zu jedem Fräsprojekt existiert immer ein Satz NC-Dateien, in dem alle Befehle zum Fertigen des gewünschten Werkstücks enthalten sind. Ist ein Projekt erfolgreich simuliert, werden die NC-Dateien unverändert für die reale Fräsbearbeitung weiterverwendet.

Konfigurationsdatei (XML)

In dieser Datei wird die virtuelle Fräsmaschine eingerichtet, indem ein unbearbeitetes Werkstück (Halbzeug) definiert und die Werkzeugmagazinbelegung der Maschine festgelegt wird.

Das folgende Beispiel zeigt eine vollständige Konfigurationsdatei, die ein quaderförmiges Rohteil definiert, das von vier Spannzangen fixiert wird. Des Weiteren werden dem Werkzeugmagazin sechs Werkzeuge zugeordnet.

```
<CNC3DMilling_Config xmlns="urn:STA-Milling" version="1.0">
  <Workpiece number="1" X="200" Y="200" Z="95" transX="0" transY="0" transZ="0"
    rotX="0" rotY="0" rotZ="1" rotAngle="0"/>
  <Fixture workpiece="1">
    <Claw X="50" Y="50" Z="50" transX="-125" transY="0" transZ="0" rotX="0"
      rotY="0" rotZ="1" rotAngle="0"/>
    <Claw X="50" Y="50" Z="50" transX="0" transY="-125" transZ="0" rotX="0"
      rotY="0" rotZ="1" rotAngle="0"/>
    <Claw X="50" Y="50" Z="50" transX="125" transY="0" transZ="0" rotX="0"
      rotY="0" rotZ="1" rotAngle="0"/>
    <Claw X="50" Y="50" Z="50" transX="0" transY="125" transZ="0" rotX="0"
      rotY="0" rotZ="1" rotAngle="0"/>
  </Fixture>
  <Magazine size="100">
    <Tool position="1" filename="t1spibo.gd"/>
    <Tool position="2" filename="t2schrupp.gd"/>
    <Tool position="3" filename="T3radius.gd"/>
    <Tool position="4" filename="T4shaft.gd"/>
    <Tool position="5" filename="t5anbohrer.gd"/>
  </Magazine>
</CNC3DMilling_Config>
```

```
<Tool position="6" filename="t6tnut.gd"/>
</Magazine>
<Spindle tool="t6tnut.gd" magazine="0"/>
</CNC3DMilling_Config>
```

Das Halbzeug wird durch das Element 'Workpiece' repräsentiert. Mit den zugehörigen Attributen lässt sich ein Quader definieren und beliebig innerhalb der Maschine positionieren.

Um ein Werkstück in der Fräsmaschine zu fixieren, bedarf es sogenannter Spannmittel, die in der Simulation stets zu Quadern vereinfacht werden. In der Konfigurationsdatei wird für jedes Spannmittel ein Element 'Claw' definiert.

Wie bereits beschrieben, müssen für die zu simulierenden Werkzeuge GD-Dateien existieren. Um das Werkzeugmagazin der virtuellen Maschine zu bestücken, werden über die Attribute 'position' und 'filename' den einzelnen Magazinplätzen GD-Dateien zugeordnet.

Das Element „Spindle“ kann genutzt werden, um zu Simulationsbeginn die Spindel mit einem Werkzeug vorzubereiten.

Milling Project (MPJ)

In einer Projektdatei wird ein komplettes Simulationsprojekt konfiguriert. Hierzu wird eine Maschine in Form einer GD-Datei, eine Konfigurationsdatei und ein NC-Programm ausgewählt. Damit sind alle nötigen Vorbereitungen getroffen und die Simulation kann durchgeführt werden.

```
<CNC3DMilling_Project xmlns="urn:STA-Milling" version="1.0">
  <Machine filename="machine.gd"/>
  <ConfigurationFile filename="sample.xml"/>
  <NCProgram filename="sample.mnc"/>
</CNC3DMilling_Project>
```

2.3 Marktvergleich von Werkzeugverwaltungen

Es gibt auf dem Markt eine Vielzahl von Werkzeugverwaltungen, die neben der eigentlichen Verwaltung von Datensätzen, eine große Vielfalt von Zusatzmodulen zur Verfügung stellen. Diese decken den kompletten Lebenszyklus eines Werkzeugs - vom Einkauf über Lagerung bis zur Verschleiß- und Qualitätskontrolle - ab.

Die verbreitetsten drei Verwaltungen dieser Art sind CimSource Professional ([CIM 03]), Tool Data Management ([Walter 03]) und WinTool ([Datos 03]).

Neben diesen Werkzeugverwaltungen gibt es viele Simulationen, PPS- oder CAD/CAM-Systeme (NC Verify, Visual CNC, TopCAM, PEPS, ...), die über eigene Werkzeugmodule verfügen. Kennzeichnend für diese Systeme ist, dass deren Werkzeugverwaltungen nur Zusatzmodule und nicht die eigentlichen Kernprogramme sind, weshalb der Funktionsumfang gegenüber reinen Werkzeugverwaltungen reduziert ist.

In Abbildung 4 wird der Leistungsumfang einiger verbreiteter Werkzeugverwaltungen verglichen:

	CIMSOURCE	TDM	WinTool	PEPS CAD/CAM*
Datenquellen	Herstellerkataloge	Walter Werkzeugkatalog	Herstellerkataloge	PEPS Werkzeugdatenbank
Verwaltung von Werkzeugaufnahmen	ja	ja	ja	ja
Verwaltung von Komplettwerkzeugen	ja	ja	ja	ja
Suchkriterien	Suche über Geometrie, Werkstoff, Normbezeichnung, Bearbeitungsaufgabe	Suche über Geometrie, Werkstoff, Normbezeichnungen, Bearbeitungsaufgabe, Sachmerkmale	Suche über Geometrie, Werkstoff, Sachmerkmale	Suche über Geometrie, Werkstoff,
Schnittwertbestimmung	ja	ja	ja	nein
Plausibilitätscheck	ja	ja	ja	nein
automatische Zeichnungserstellung	ja	ja	ja	nein
Exportfunktionen	dxp	dxp, iges, vdaps	dxp	nein
Schnittstellen zu anderen Systemen	diverse NC- und PPS-Systeme	diverse NC- und PPS-Systeme	diverse NC- und PPS-Systeme	keine
Zusatzmodule				

	CIMSOURCE	TDM	WinTool	PEPS CAD/CAM*
Beschaffungsmodul (Anfragen, Bestellungen, Ersatzteile, ...)	ja	ja	ja	nein
Lagermodul (, Bestellungen, Inventur, Sortimentsbereinigung, ...)	ja	ja	ja	nein
Voreinstellmodul (Schnittstellen zu Voreinstellgeräten, Korrekturwertermittlung)	ja	ja	ja	nein
QS-Modul (DIN-ISO 9000-9004, Lieferantenbewertung)	ja	ja	nein	nein
Barcodemodul (Identnummer-, Kostenstellen Erfassung)	nein	ja	nein	nein

ABBILDUNG 4. Vergleich verschiedener Werkzeugverwaltungen

* die Angaben beziehen sich nur auf das Werkzeugmodul

Wie man der Tabelle entnehmen kann, beschränkt sich das PEPS-Werkzeugmodul auf die reine Verwaltung von Werkzeugen. Da die STA-Werkzeugverwaltung ebenfalls ein Zusatzmodul werden soll, wird der Schwerpunkt auch dort bei der eigentlichen Verwaltung von Aufnahmen, Einzel- und Komplettwerkzeugen liegen. Es sind jedoch Zusatzfunktionen, wie Plausibilitätsprüfungen oder automatische 3D-Zeichnungserstellung geplant. Eine genaue Spezifikation des Systems folgt in Kapitel 3.

3 Problembeschreibung und Lösung

Aufgabenstellung für diese Diplomarbeit ist der Entwurf und die Implementierung einer Werkzeugverwaltung für die Frässimulation 'CNC Milling'. Um das System vorerst einmal grob zu beschreiben, folgt eine Auflistung von grundlegenden Anforderungen:

- Es soll ein System zur Klassifizierung der zu verwaltenden Werkzeuge entwickelt werden. Es müssen - mit einem einheitlichen System - alle denkbaren Fräs- und Bohrwerkzeuge erfasst werden können.
- Da nicht nur Werkzeuge, sondern auch die zugehörigen Werkzeugaufnahmen verwaltet werden sollen, muss auch eine Klassifizierung für die Aufnahmen entworfen werden.
- Zur Beschreibung eines Werkzeugs bzw. einer Aufnahme sollen nicht nur die reinen Geometriedaten verwendet werden, sondern auch beschreibende und technologische Parameter, wie zum Beispiel Kommentare, Schneiden- und Schaftausführungen, Schneidstoffe u.s.w..
- Es soll eine Bedienoberfläche erstellt werden, die alle nötigen Aktionen ermöglicht und eine graphische Vorschau für Werkzeuge und Aufnahmen bietet.
- Um die Werkzeugverwaltung für die Simulation nutzbar zu machen, sollen aus gespeicherten Komplettwerkzeugen Geometriedaten-Dateien erzeugt werden können, die direkt von der Simulation eingelesen werden können.

3.1 Entwicklung eines Klassifizierungssystems

In diesem Abschnitt wird ein System zur Klassifizierung der Fräswerkzeuge und Aufnahmen entwickelt. Dafür muss zum einen eine Methodik zur Einordnung der Werkzeuge und Aufnahmen geschaffen werden. Zum anderen muss eine Auswahl von Parametern getroffen werden, die zur Beschreibung der verwalteten Komponenten dient. Es muss dabei sicher gestellt werden, dass die gewählten Parameter für alle Werkzeug- und Aufnahmetypen verwendet werden können. Es darf also nicht vorkommen, dass eine Komponente nicht in die Werkzeugverwaltung aufgenommen werden kann, weil es nicht durch die zur Verfügung stehenden Parameter beschrieben werden kann.

Abbildung 5 zeigt einige Beispiele für Fräswerkzeuge.



ABBILDUNG 5. Verschiedene Fräswerkzeuge; links: Gewindefräser; mitte: Versenk- und Anbohrfräser; rechts: diverse Schaftfräser und ein Walzenkopffräser
 Quelle: Gühring oHG (www.guehring.de)

3.1.1 Entwerfen einer Methodik zur Klassifizierung

Eine normierte Klassifizierung der meisten verfügbaren Werkzeuge ist in [DIN 92] zu finden. Das DIN-System ist sehr ausführlich und beschreibt jede Werkzeugklasse im Detail (samt Schneidenwinkeln, Toleranzen, Oberflächengüten, u.s.w.).

Es existieren bereits andere Systeme zur Klassifizierung von Werkzeugen, welche - je nach Verwendungszweck - verschiedene Konzepte verfolgen und die Schwerpunkte unterschiedlich setzen. Die in diesen Systemen verwendeten Parameter stellen i.d.R. eine Untermenge der DIN-Parameter dar.

Eine Gemeinsamkeit dieser Systeme besteht jedoch in der Methodik der Werkzeugeinordnung. Sie alle sortieren die Werkzeuge nach sogenannten Werkzeugklassen, von denen in [DIN 92] über 60 Stück definiert sind, zum Beispiel Kugelkopffräser, T-Nutenfräser, Walzenstrinfräser, Messerkopffräser und viele mehr.

Die Werkzeugaufnahmen werden nach dem Typ des Werkzeugadapters geordnet, das ist der Teil einer Aufnahme, in (bzw. an) dem das Werkzeug fixiert wird - z.B. ein Bohrfutter oder eine Aufnahme für Zylinderschäfte. Ein zweites Kriterium zum Klassifizieren von Aufnahmen ist der genormte Aufnahmekegel. Dieser dient dazu, die Werkzeugaufnahme in der Antriebsspindel der Fräsmaschine zu fixieren. Einige Standards für Aufnahmekegel sind: DIN 2080, DIN 69871, HSK, Capto, u.s.w..

Abbildung 6 zeigt einen Kugelkopffräser (schwarzer Schaft mit roter Schneide) in einer Aufnahme

nach DIN 69871 (dazu gehört der Kegel und der angrenzende Aufnahmering) mit einem Fräser-
spannfutter als Werkzeugaufnahme (grüner Teil rechts vom Aufnahmering).



ABBILDUNG 6. Kugelkopffräser mit Aufnahme aus der Simulation 'CNC Milling'
Quelle: STA GmbH

Dieses Klassifizierungssystem ist sehr weit verbreitet und wird deshalb für die zu erstellende Werkzeugverwaltung unverändert übernommen. Damit ist auch gewährleistet, dass später einmal Schnittstellen zu bestehenden Werkzeugverwaltungen geschaffen werden können.

3.1.2 Richtlinien zur Auswahl der Parameter

Für die Auswahl der Parameter, mit denen die Werkzeuge und Aufnahmen beschrieben werden, gibt es, wie in Abschnitt 3.1.1 erwähnt, umfangreiche Richtlinien in [DIN 92]. Diese sind jedoch nur bedingt für die Werkzeugverwaltung geeignet, da für die unterschiedlichen Klassen oft sehr verschiedene Parameter zur Beschreibung verwendet werden. In der Werkzeugverwaltung soll aber ein einheitliches System verwendet werden, was letztendlich die automatische Generierung von Werkzeugmodellen für die Simulation ermöglicht. Es gilt also, eine Parametermenge zu finden, die möglichst DIN-konform ist und gleichzeitig den Anforderungen der Simulation gerecht wird. Im folgenden werden einige Grundanforderungen formuliert, die als Richtlinien zur Auswahl der letztendlich verwendeten Parameter dienen sollen.

Verwendungszweck

Der hauptsächlich auftretende Anwendungsfall wird das Erzeugen von Geometriedaten-Dateien aus Komplettwerkzeugen sein, die in der Werkzeugverwaltung hinterlegt sind. Hierfür werden lediglich die geometrischen Werkzeugdaten, also Längen, Durchmesser, Radien, u.s.w. benötigt.

Detaillierungsgrad der geometrischen Parameter

Grundsätzlich sollte ein geringer Detaillierungsgrad angestrebt werden, um Rechenzeit und Verwaltungsaufwand zu sparen. Während eines Simulationsvorgangs wird zur Berechnung von Materialentnahme und zur Kollisionsprüfung nur die Kontur und Lage der Schneide berücksichtigt, während Werkzeugschaft und Aufnahme nicht beachtet werden. Deshalb sollte die Schneide stets exakt

beschrieben sein, während Aufnahme und Schaft vereinfacht dargestellt werden können. Da eine ansprechende grafische Darstellung jedoch ein wesentliches Merkmal der Simulation ist, gilt es, einen Kompromiss zwischen den dargestellten Details und dem Verarbeitungsaufwand zu finden. Für die Klassifizierung bedeutet dies, dass alle nicht dargestellten Details auch nicht in das System aufgenommen werden.

Auswahl der zusätzlichen Parameter

Die 'zusätzlichen Parameter' sind alle Angaben zu einem Werkzeug, die nicht für die direkte Verwendung in der Simulation nötig sind. Dazu gehören kurze Beschreibungen, die das Auffinden eines Werkzeugs beim Suchen in Listen erleichtern oder Kommentare, die Hinweise auf besondere Eigenarten oder Verwendungszwecke geben. Weitere Informationen können Herstellerdaten, Bestellinformationen, buchhalterische Daten, logistische Angaben und Ähnliches sein. Da jedoch die Hauptaufgabe des Systems die Bereitstellung von Datensätzen ist, wird auf Parameter wie 'Inventarnummer' oder 'Lagerort' verzichtet.

Parameter zur Schnittwertermittlung

Schnittwerte sind Parameter, die zu einer bestimmten Fräsaufgabe (zum Beispiel dem Bearbeiten von Aluminium mit einem Schaftfräser aus Hartmetall) gehören. Beispiele für Schnittwerte sind Spindeldrehzahl, Vorschub, Schnittgeschwindigkeit u.Ä.. Sie lassen sich für jeden Bearbeitungsfall berechnen - vorausgesetzt, man verfügt über die nötigen Eckdaten. Das sind zum einen die verwendeten Werkstoffe für Werkstück und Werkzeugschneide, zum anderen geometrische Parameter wie Schneidendurchmesser, Schneidenzahl, Eingriffstiefe, Eingriffsbreite u.Ä..

Große Werkzeugverwaltungssysteme verfügen über sogenannte Expertenmodule, die in der Lage sind, aus den oben genannten Eckdaten und herstellerspezifischem Spezialwissen die Schnittwerte für eine Fräsbearbeitung zu ermitteln.

Für die zu erstellende Werkzeugverwaltung ist kein solches Expertenmodul vorgesehen; es sollte aber dennoch möglich sein, aus den zur Verfügung stehenden Daten, die nötigen Schnittwerte zu ermitteln. Die dafür nötigen Parameter sind Schneidenwerkstoff, Schneidenzahl und Schneidengeometrie. Werkstückwerkstoff, Eingriffstiefe und -breite sind vom Anwendungsfall abhängig und werden daher nicht gespeichert.

3.1.3 Erläuterung der Parameter

Nach den Vorgesandten zur Auswahl der Parameter folgt nun eine kurze Beschreibung der letztendlich ausgewählten Größen. Die Parameter sind dabei in drei Kategorien geordnet:

- Werkzeuge Ein Werkzeug besteht aus einer Schneide und dem zugehörigen Halter, was ein Schaft oder eine Aufnahmebohrung sein kann. In Abbildung 7 sind ein Kugelpkopfräser und ein Walzenstirnfräser dargestellt.

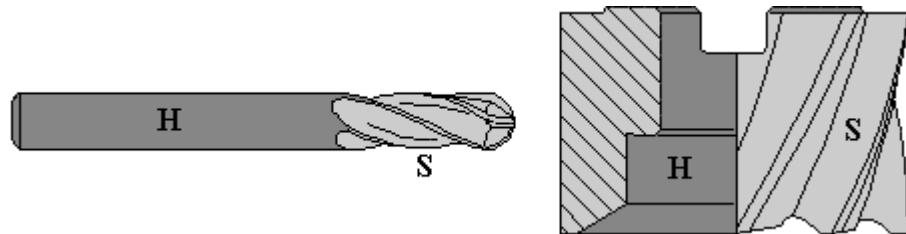


ABBILDUNG 7. Werkzeuge (H=Halter (kann ein Schaft oder eine Aufnahmebohrung sein); S=Schneide)

- Aufnahmen Eine Aufnahme ist ein Zwischenstück, das benötigt wird, um ein Werkzeug mit einer Maschinenspindel zu verbinden. Abbildung 8 zeigt links eine Aufnahme nach DIN 69871 mit Fräuserspannfutter und rechts eine Aufnahme nach DIN 2080 mit Aufsteckfräsdorn; Beispiele für jeweils passende Werkzeuge sind in Abb. 7 dargestellt.

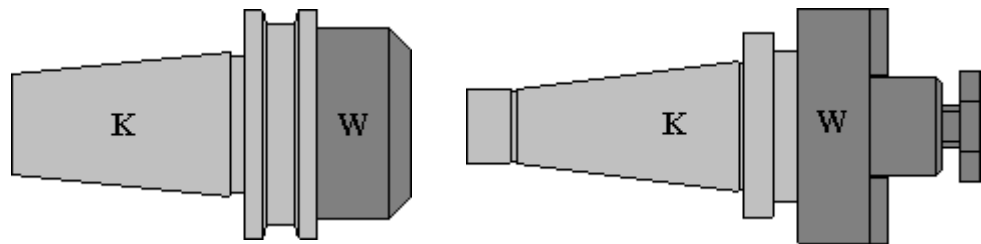


ABBILDUNG 8. Werkzeugaufnahmen (K=genormter Aufnahmekegel, W=Werkzeugadapter)

- Komplettwerkzeuge Ein Komplettwerkzeug ist eine Kombination aus Werkzeug und Aufnahme. Abbildung 9 zeigt einen Winkelfräser in einer Aufnahme nach DIN 69871

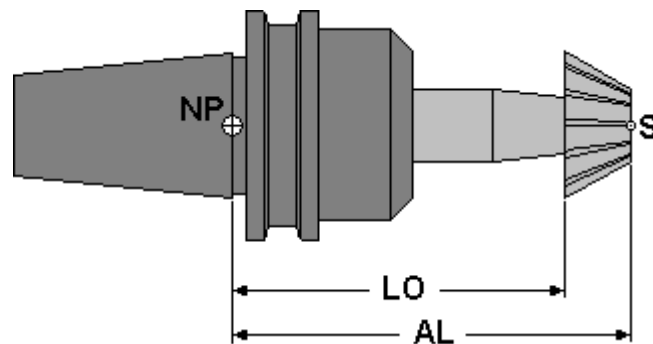


ABBILDUNG 9. Komplettwerkzeug (NP=Werkzeugnullpunkt der Maschine; LO=Längenoffset; AL=Ausspannlänge; S=Werkzeugspitze)

Parameter für Werkzeuge:

Werkzeugklasse

Die Werkzeugklasse legt fest, welche geometrischen Parameter zu einem Werkzeug gehören und wie es grundsätzlich in die Werkzeugverwaltung eingeordnet wird (vgl. Abschnitt 3.1.1).

Bsp.: Schaftfräser: - Schneidendurchmesser
- Schneidenlänge
- Schaftdurchmesser
- Schaftlänge
- Eckenradius

Werkzeugbeschreibung

Die Werkzeugbeschreibung sollte möglichst kurz gehalten werden, sie ist der wesentliche Suchparameter zur Auswahl eines Werkzeugs aus einer Liste.

Bsp.: „Schlichtfräser8“
„TNutenfräser20x8“

Werkzeugkommentar

Hier können Besonderheiten aller Art zu einem Werkzeug eingetragen werden.

Bsp.: „extra lang“
„rechtsschneidend“
„zentrumschneidend“

Anzahl der Schneiden

Gibt die Anzahl der Schneiden, bzw. Schneidplatten eines Werkzeugs wieder.

Halterausführung

Ein Halter ist der Teil des Werkzeugs, der keine Schneide ist. Das kann ein Schaft oder eine Aufnahmebohrung sein (siehe Abb.7). Durch diesen Parameter lässt sich erkennen, welche Aufnahme für das Werkzeug verwendet werden kann.

Bsp.: „Zylinderschaft mit Spannfläche“
„Fräsdornaufnahme mit Quernut“

Schneidenausführung

Eine Schneide (siehe Abb.7) kann über sehr viele verschiedene Ausführungen verfügen. Merkmale solcher Ausführungen sind z.B. Helix (Schneidensteigung), Spanwinkel, Freiwinkel, Art der Verzahnung, Art der Wendeschneidplatten, u.s.w.. Mit diesem Parameter können die einzelnen Merkmale textuell beschrieben werden.

Bsp.: „Typ NR“
„Helix 30°, Spanwinkel 14°“
„Schruppverzahnung mit Spanteiler“

Geometrische Abmessungen

Art und Anzahl der geometrischen Parameter werden durch die Werkzeugklasse (siehe oben) definiert.

Werkzeughersteller

Gibt den Werkzeughersteller an.

Bsp.: „Coromant“
„Biax“

Schneidenmaterial

Gibt den Grundwerkstoff an, aus dem die Schneide besteht; oft wird sie zusätzlich beschichtet (siehe unten).

Bsp.: „HSS“
„VHM“

Schneidenbeschichtung

Gibt (wenn vorhanden) den Beschichtungswerkstoff der Schneide an.

Bsp.: „TiN“
„TiCN“

Kommentar zum Schneidenwerkstoff

Hier kann ein Hinweis auf die Anwendungsmöglichkeiten des Werkstoffs angegeben werden.

Bsp.: „Nicht für eisenhaltige Werkstoffe geeignet“
„Für zähe Werkstoffe und HSC-Bearbeitung“

Parameter einer Aufnahme:

Aufnahmebeschreibung

Die Beschreibung sollte möglichst kurz gehalten werden und dient dem Auffinden einer Aufnahme aus einer Auswahlliste.

Bsp.: „Bohrfutter1-13 SK30 DIN2080“
„Frässpännfutter08 SK40 DIN69871“

Aufnahmezeichnung

Eine technische Zeichnung, die alle relevanten Maße für eine Aufnahme angibt. Die Zeichnung ist unbedingt erforderlich, da sie als einziges Objekt die Bedeutung der geometrischen Aufnahmeparameter erklärt.

Geometrische Abmessungen

Art und Anzahl der geometrischen Abmessungen werden durch den Werkzeugaufnahmetyp (siehe unten) definiert und durch die Aufnahmezeichnung (siehe oben) erklärt.

Hersteller

Gibt den Hersteller der Aufnahme an.

Kegeltyp

Beschreibt den Kegeltyp der Aufnahme (siehe Abb.8). Alle Kegeltypen sind genormt und müssen deshalb nicht durch geometrische Parameter beschrieben werden.

Bsp.: „SK 30“
 „SK 50“

Kegelnorm

Ein Kegel (siehe 'Kegeltyp') kann nach verschiedenen Normen gefertigt sein, die hier angegeben werden können.

Bsp.: „DIN 2080“
 „DIN 69871“

Werkzeugaufnahmetyp

Der Werkzeugaufnahmetyp (siehe Abb.8) ist der werkstückzugewandte Teil einer Aufnahme, in dem ein Werkzeug fixiert werden kann.

Bsp.: „Bohrfutter“
 „Fraeserspannfutter“

Kommentar zur Werkzeugaufnahme

Ein Hinweis zu den Verwendungsmöglichkeiten eines Werkzeugaufnahmetyps (siehe oben).

Bsp.: „für Werkzeuge mit zylindrischem Schaft“

Parameter eines Komplettwerkzeugs:

Ausspannlänge

Der Abstand zwischen dem Werkzeugnullpunkt der Maschine und der Werkzeugspitze (siehe Abb.9).

Radiuskorrekturwert

In einer NC-Datei ist die zu erzeugende Werkstückkontur in Form von sehr vielen Punkten angegeben. Eine NC-Steuerung lässt das Fräswerkzeug genau an diesen Punkten entlangfahren. Ohne zusätzliche Maßnahmen würde aber die dadurch erzeugte Werkstückkontur genau um den Fräserradius zu klein sein, da die Steuerung immer den Werkzeugmittelpunkt verfährt. Für eine Fräsbahnkorrektur wird deshalb ein Radiuskorrekturwert angegeben, um den das Werkzeug äquidistant zur tatsächlichen Fräsbahn verfahren werden soll. Sehr oft ist der Korrekturwert gleich dem Schneidradius, wie Abbildung 10 verdeutlicht.

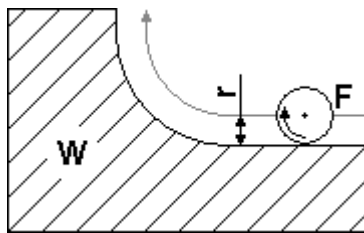


ABBILDUNG 10. Radiuskorrektur (W=Werkstück; F=Fräser; r=Radiuskorrekturwert)

Längenoffset

Der Abstand vom Werkzeugnullpunkt der Maschine bis dem Punkt an der Werkzeugschneide, der für die Fräsbahnkorrektur (vgl. 'Radiuskorrekturwert') verwendet werden soll (vgl. Abb.9). Oft ist der Längenoffset gleich der Ausspannlänge.

3.2 Spezifikation der Software

Nachdem im vorangegangenen Abschnitt eine Klassifizierung für die Werkzeugverwaltung festgelegt wurde, folgt nun eine Beschreibung der Aufgaben, die von der Benutzeroberfläche erfüllt werden sollen.

3.2.1 Anwendungsfalldiagramm

Abbildung 11 zeigt die grundlegenden Anwendungsfälle, die in der Werkzeugverwaltung auftreten können. Eine Beschreibung der hier verwendeten Syntax liefert z.B. [Booch et al. 99]. Es werden dabei zwei Rollen unterschieden, die ein Benutzer einnehmen kann. Sollen Datensätze verwaltet, also neu erstellt, geändert oder gelöscht werden, so fällt dem Benutzer die Rolle des Administrators zu. Dieser Fall tritt auf, wenn die Werkzeugverwaltung neu installiert wurde und der tatsächliche Werkzeugbestand einer Werkstatt in die Datenbank eingegeben wird oder wenn der Inhalt der Datenbank aktualisiert und mit dem tatsächlichen Werkzeugbestand abgeglichen wird.

Der häufiger auftretende Anwendungsfall wird jedoch das Erstellen von GD-Dateien für ein neues Fräsprojekt sein. Dies kann mittels Komplettwerkzeugen geschehen, die bereits in der Datenbank abgelegt sind oder aus einem Werkzeug und einer Aufnahme neu zusammengesetzt werden.

Es ist theoretisch auch möglich, alle existierenden Werkzeuge permanent als GD-Dateien vorliegen zu haben, was allerdings wenig sinnvoll wäre, da diese Daten redundant zu denen in der Datenbank vorlägen. Des weiteren wird die Übersichtlichkeit verloren gehen, wenn mehrere hundert Werkzeuge als GD-Dateien gespeichert werden, da die Benennungen der Dateien normalerweise keine eindeutigen Rückschlüsse auf die dargestellten Werkzeuge zulassen (z.B.: „Schaftfräser10.gd“).

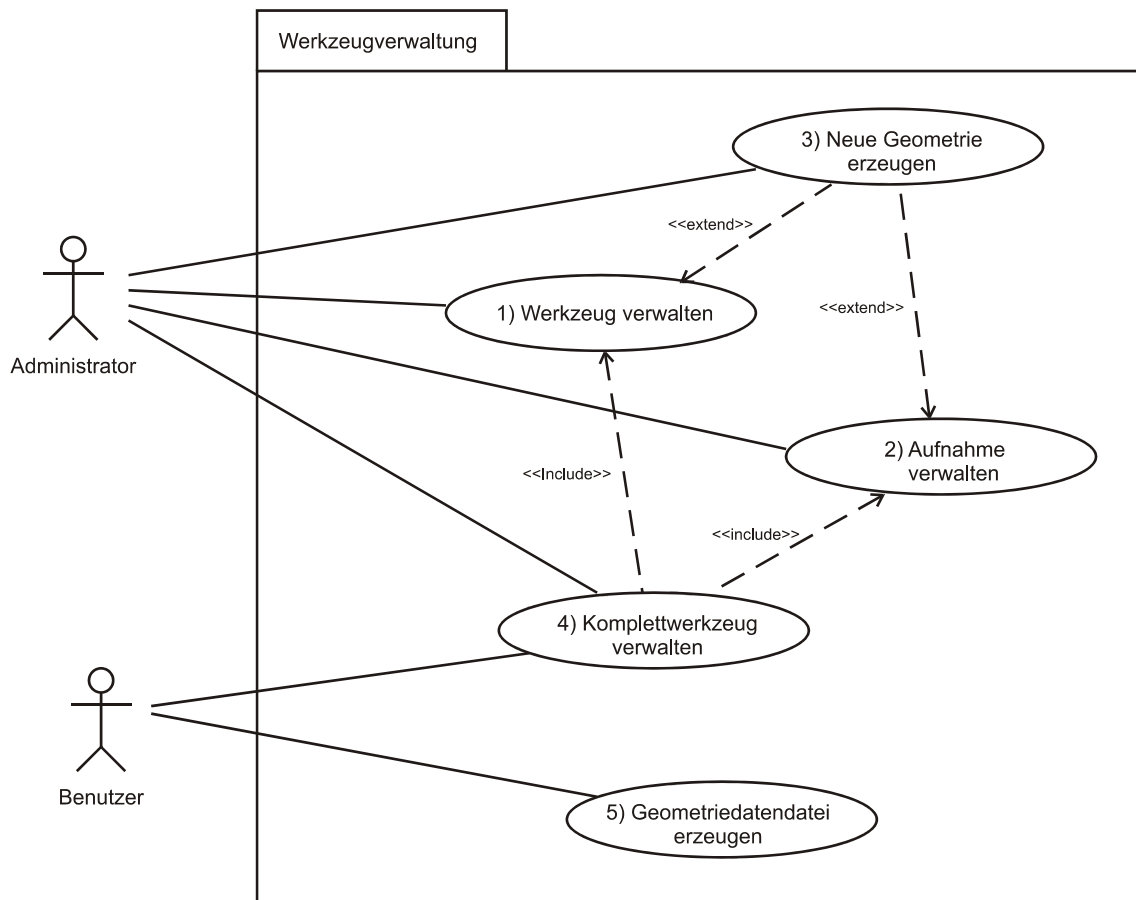


ABBILDUNG 11. Anwendungsfalldiagramm

Zur näheren Erläuterung der dargestellten Fälle folgt jeweils eine textuelle Beschreibung, wobei einige Anwendungsfälle (AF) in verschiedene Szenarien unterteilt sind. Die Notation richtet sich nach [Oestereich 98].

AF 1.1) Werkzeug neu anlegen

Vorbedingungen:

Schreibzugriff auf eine Datenbank

Invarianten:

Datenintegrität muss gewährleistet sein

Ablaufbeschreibung:

1. Auswahl und Eingabe der Werkzeugparameter:

- Werkzeugklasse
- Werkzeugbeschreibung

- Werkzeugkommentar (optional)
- Ausführung des Halters (optional)
- Ausführung der Schneide (optional)
- Werkzeughersteller
- Schneidenwerkstoff, Schneidenbeschichtung
- Werkzeugmaße
- Schneidenzahl
- Technische Zeichnung (wenn eine Andere als die Standardzeichnung gewünscht ist)

2. Werkzeug speichern

Ausnahmen, Fehlersituationen:

Vor dem Speichern wird geprüft, ob die eingegebenen Parameter vollständig und gültig sind. Das Anlegen des neuen Werkzeugs erfolgt nur, wenn kein Fehler gefunden wurde.

AF 1.2) Werkzeugvariante erstellen

Vorbedingungen:

Lese-/Schreibzugriff auf eine Datenbank

Invarianten:

Datenintegrität muss gewährleistet sein

Ablaufbeschreibung:

1. Werkzeug auswählen
2. Änderungen vornehmen
3. Werkzeug speichern

Ausnahmen, Fehlersituationen:

Vor dem Speichern wird geprüft, ob die eingegebenen Parameter vollständig und gültig sind. Das Anlegen des neuen Werkzeugs erfolgt nur, wenn kein Fehler gefunden wurde.

AF 1.3) Werkzeug ändern

Vorbedingungen:

Lese-/Schreibzugriff auf eine Datenbank

Invarianten:

Datenintegrität muss gewährleistet sein

Ablaufbeschreibung:

1. Werkzeug auswählen
2. Änderungen eingeben
3. Änderungen bestätigen

Ausnahmen, Fehlersituationen:

Nach dem Bestätigen wird geprüft, ob die eingegebenen Parameter vollständig und gültig sind.
Das Übernehmen der Änderungen erfolgt nur, wenn kein Fehler gefunden wurde.

AF 1.4) Werkzeug löschen

Vorbedingungen:

Lese-/Schreibzugriff auf eine Datenbank

Es muss ein Werkzeug ausgewählt sein

Invarianten:

Datenintegrität muss gewährleistet sein

Ablaufbeschreibung:

1. Werkzeug auswählen
2. Werkzeug löschen

Ausnahmen, Fehlersituationen:

Vor dem Löschen wird geprüft, ob das Werkzeug in einem Komplettwerkzeug verwendet wird. Ist das der Fall, wird das Löschen nicht durchgeführt.

AF 2.1) Aufnahme neu anlegen

Vorbedingungen:

Schreibzugriff auf eine Datenbank

Invarianten:

Datenintegrität muss gewährleistet sein

Ablaufbeschreibung:

1. Auswahl und Eingabe der Parameter:
 - Beschreibung der Aufnahme

- Typ der Werkzeugaufnahme
- Kegeltyp
- Hersteller der Aufnahme
- Technische Zeichnung (wenn eine Andere als die Standardzeichnung gewünscht ist)
- Geometrische Abmessungen

2. Aufnahme speichern

Ausnahmen, Fehlersituationen:

Vor dem Speichern wird geprüft, ob die eingegebenen Parameter vollständig und gültig sind. Das Anlegen der neuen Aufnahme erfolgt nur, wenn kein Fehler gefunden wurde.

AF 2.2) Variante einer Aufnahme erstellen

Vorbedingungen:

Lese-/Schreibzugriff auf eine Datenbank

Invarianten:

Datenintegrität muss gewährleistet sein

Ablaufbeschreibung:

1. Aufnahme auswählen
2. Änderungen vornehmen
3. Aufnahme speichern

Ausnahmen, Fehlersituationen:

Vor dem Speichern wird geprüft, ob die eingegebenen Parameter vollständig und gültig sind. Das Anlegen der neuen Aufnahme erfolgt nur, wenn kein Fehler gefunden wurde.

AF 2.3) Aufnahme ändern

Vorbedingungen:

Lese-/Schreibzugriff auf eine Datenbank

Invarianten:

Datenintegrität muss gewährleistet sein

Ablaufbeschreibung:

1. Aufnahme auswählen

2. Änderungen eingegeben

3. Änderungen bestätigen

Ausnahmen, Fehlersituationen:

Nach dem Bestätigen wird geprüft, ob die eingegebenen Parameter vollständig und gültig sind. Das Übernehmen der Änderungen erfolgt nur, wenn kein Fehler gefunden wurde.

AF 2.4) Aufnahme löschen

Vorbedingungen:

Lese-/Schreibzugriff auf eine Datenbank

Invarianten:

Datenintegrität muss gewährleistet sein

Ablaufbeschreibung:

1. Aufnahme auswählen

2. Aufnahme löschen

Ausnahmen, Fehlersituationen:

Vor dem Löschen wird geprüft, ob die Aufnahme in einem Komplettwerkzeug verwendet wird. Ist das der Fall, wird das Löschen nicht durchgeführt.

AF 3) Neue Geometrie erzeugen

Vorbedingungen:

Schreibzugriff auf eine Datenbank

Ablaufbeschreibung:

1. Neue Werkzeugklasse in der Datenbank anlegen

2. Beschreibende Zeichnung bereitstellen

3. Bildungsvorschriften für neues GD-Objekt erzeugen

4. Den Bildungsvorschriften Regeln zur Wertprüfung hinzufügen

Varianten:

1a) Handelt es sich um eine Aufnahme, so muss statt der Werkzeugklasse ein neuer Werkzeugadapter und / oder ein neuer Aufnahmekegeltyp erstellt werden.

AF 4.1) Komplettwerkzeug neu anlegen

Vorbedingungen:

Lese-/Schreibzugriff auf eine Datenbank

Invarianten:

Datenintegrität muss gewährleistet sein

Ablaufbeschreibung:

1. Werkzeug auswählen
2. Aufnahme auswählen
3. Parameter eingeben:
 - Ausspannlänge
 - Längenoffset
 - Durchmesseroffset
4. 3D-Geometrie erstellen (optional)
5. Komplettwerkzeug speichern

Ausnahmen, Fehlersituationen:

Vor dem Speichern wird geprüft, ob die eingegebenen Parameter vollständig und gültig sind. Das Anlegen des neuen Komplettwerkzeugs erfolgt nur, wenn kein Fehler gefunden wurde.

Vor dem Erzeugen der 3D-Geometrie wird eine Werteprüfung der Parameter durchgeführt. Die 3D-Geometrie wird nur erzeugt, wenn keine Fehler gefunden wurden.

Die 3D-Geometrie kann nur erzeugt werden, wenn alle erforderlichen Erstellungsvorschriften zur Verfügung stehen.

AF 4.2) Variante eines Komplettwerkzeugs erstellen

Vorbedingungen:

Lese-/Schreibzugriff auf eine Datenbank

Invarianten:

Datenintegrität muss gewährleistet sein

Ablaufbeschreibung:

1. Komplettwerkzeug auswählen
2. Änderungen vornehmen
3. Komplettwerkzeug speichern

Ausnahmen, Fehlersituationen:

Vor dem Speichern wird geprüft, ob die eingegebenen Parameter vollständig und gültig sind. Das Anlegen des neuen Komplettwerkzeugs erfolgt nur, wenn kein Fehler gefunden wurde.

AF 4.3) Komplettwerkzeug ändern

Vorbedingungen:

Lese-/Schreibzugriff auf eine Datenbank

Invarianten:

Datenintegrität muss gewährleistet sein

Ablaufbeschreibung:

1. Komplettwerkzeug auswählen
2. Änderungen eingeben
3. Änderungen bestätigen

Ausnahmen, Fehlersituationen:

Nach dem Bestätigen wird geprüft, ob die eingegebenen Parameter vollständig und gültig sind. Das Übernehmen der Änderungen erfolgt nur, wenn kein Fehler gefunden wurde.

AF 4.4) Komplettwerkzeug löschen

Vorbedingungen:

Lese-/Schreibzugriff auf eine Datenbank

Invarianten:

Datenintegrität muss gewährleistet sein

Ablaufbeschreibung:

1. Komplettwerkzeug auswählen

2. Komplettwerkzeug löschen (die einzelnen Komponenten, also Aufnahme und Werkzeug bleiben erhalten)

AF 5) Geometriedatei erzeugen

Vorbedingungen:

Lese-/Schreibzugriff auf eine Datenbank;

Es müssen Komplettwerkzeuge in der Datenbank bereitliegen

Ablaufbeschreibung:

1. Komplettwerkzeug auswählen
2. Geometriedatei erzeugen

Ausnahmen, Fehlersituationen:

Vor dem Erzeugen der 3D-Geometrie wird eine Werteprüfung der Parameter durchgeführt. Die 3D-Geometrie wird nur erzeugt, wenn keine Fehler gefunden wurden.

Die 3D-Geometrie kann nur erzeugt werden, wenn alle erforderlichen Erstellungsvorschriften zur Verfügung stehen.

3.2.2 Generieren von GD-Dateien aus Datensätzen

Dreh- und Angelpunkt der Werkzeugverwaltung ist die automatische Erstellung von GD-Dateien aus Datensätzen. Da in den GD-Dateien keine zusätzlichen Informationen wie Schneidstoffe, Hersteller, u.s.w. gespeichert werden, sind allein die geometrischen Parameter (Werkzeug- und Aufnahmemasse, Längensoffset, Radiuskorrektur und Ausspannlänge) bedeutsam. Für jeden Werkzeug- und Aufnahmetyp sind die Parameter in ihrer Zahl und Bedeutung unterschiedlich und jeder Typ hat unterschiedliche Bildungsvorschriften für die 3D-Konturen.

Da es jederzeit möglich sein muss, neue Aufnahme- und Werkzeugtypen zu erstellen, ist es nötig, den Quellcode zur Erstellung der GD-Dateien aus dem Hauptprogramm auszulagern. Andernfalls müsste nach jedem Hinzufügen eines Typs eine neue Programmversion kompiliert und an die Kunden verteilt werden.

Zu den Bildungsvorschriften gehören neben den Angaben zur Geometrie auch Prüfredeln, mit denen die Werte vor dem Erstellen auf ihre Plausibilität geprüft werden können. Da auch diese Regeln typspezifisch sind, müssen sie ebenfalls aus dem Hauptprogramm ausgelagert werden. Die Implementierung der GD-Dateierzeugung wird in Kapitel 4 erläutert.

4 Implementierung

4.1 Auswahl der Hilfsmittel

Programmiersprache und Entwicklungsumgebung

Zur Entwicklung der Werkzeugverwaltung wird die Programmiersprache C++ benutzt, da bei der STA GmbH alle Anwendungen hiermit entwickelt werden und ausreichend Lizenzen für die Entwicklungsumgebung „Visual Studio 6.0“ zur Verfügung stehen. Die Benutzeroberfläche wird mit Hilfe der MFC 6.0 (Microsoft Foundation Class Library) erstellt.

Datenbank

Das Speichern der Datensätze in einem Dateisystem ist nicht zweckmäßig, da bei einer professionellen Anwendung des Systems mehrere hundert Werkzeuge in der Verwaltung zu erwarten sind. Aus diesem Grund werden die Daten in einer Datenbank gehalten.

Da man nicht erwarten kann, dass ein Kunde sich eine große und vor allem teure Datenbank wie z.B. Oracle für die Werkzeugverwaltung beschafft, gilt es, ein möglichst preisgünstiges System zu wählen.

Der Datenbankserver MySQL ist kostenlos als General Public License (GPL) verfügbar. Er ist weltweit verbreitet und ausführlich im Internet ([MySQL 03]) dokumentiert, so dass weder Literatur noch Software für die Benutzung gekauft werden müssen. Bei der Installation der Werkzeugverwaltung auf mehreren, miteinander vernetzten Computern genügt es, den MySQL-Server auf einem Computer zu installieren. Die restlichen PCs können über das Netzwerk mit dem Server kommunizieren.

Um die Benutzeroberfläche nicht ausschließlich für MySQL nutzbar zu machen, wird die Datenbank über eine ODBC-Schnittstelle (Open Database Connectivity) angesprochen. Diese liegt zwischen der Bedienungsoberfläche und der Datenbank und ermöglicht es, jedes Datenbanksystem - für das es ODBC-Treiber gibt - mit der Werkzeugverwaltung zu verwenden (die Werkzeugverwaltung wurde bereits mit der Datenbank MS Access getestet). Falls ein Kunde bereits eine andere Datenbank als MySQL verwendet, kann er diese also weiter benutzen - vorausgesetzt, die benötigten Werkzeuga-

bellens sind für das entsprechende System vorhanden. Abbildung 12 veranschaulicht die Zusammenarbeit der verschiedenen Komponenten.

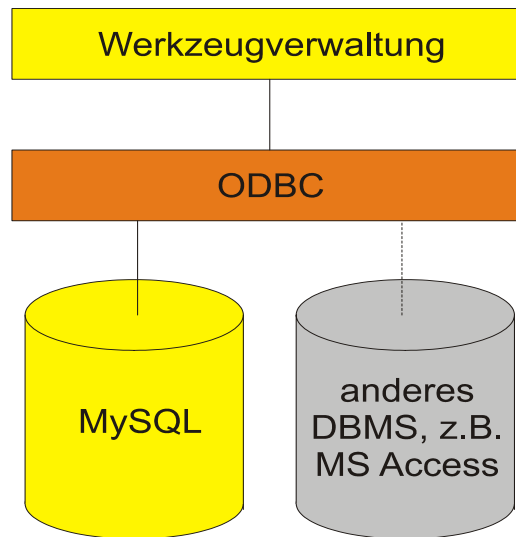


ABBILDUNG 12. Datenbankkommunikation über ODBC

Skripte

Wie bereits in Kapitel 2 beschrieben, verwendet die Simulation Geometriedaten-Dateien, um dreidimensionale Objekte darzustellen. Die Werkzeugverwaltung kann diese GD-Dateien automatisch aus gegebenen Parametern erstellen, allerdings existieren für jede Werkzeug- und Aufnahmeklasse unterschiedliche Parameter. Da es die Möglichkeit zur Erstellung neuer Werkzeugklassen geben soll, macht es Sinn, die Bildungsvorschriften für die einzelnen GD-Objekte aus dem Quellcode auszulagern.

Aus diesem Grund existiert für jede Werkzeug- und Aufnahmeklasse ein eigenes Tcl-Skript mit Bildungsvorschriften und Prüfregele. Tcl ist eine frei verfügbare Skriptsprache, mit der Zeichenketten auf besonders einfache Weise manipuliert und formatiert werden können. [Ousterhout 96] bietet eine umfassende Referenz. Zum Einbinden der Skripte in das Hauptprogramm steht ein Tcl-Interpreter zur Verfügung. Eine detaillierte Beschreibung der Skripte folgt in Abschnitt 4.4.

Plattformen

Die Werkzeugverwaltung ist für das Betriebssystem MS Windows geschrieben und wurde unter Windows98, Windows2000 und Windows XP getestet.

4.2 Klassendiagramme

Zur Erläuterung der Programmstruktur folgen zwei Klassendiagramme in UML 1.3 Syntax. Da die Diagramme einige Implementationsdetails wie Typ- und Eigenschaftsangaben zu Attributen, sowie Angaben über die Navigierbarkeit von Assoziationen enthalten, sind sie in der Implementationsbeschreibung und nicht bei der Konzeption eingeordnet (siehe Kapitel 4.2 in [Fowler & Scott 2000]).

4.2.1 Datenmodell

Abbildung 13 stellt den Aufbau der MySQL-Tabellen dar und zeigt alle verwendeten Attribute mit ihren Eigenschaften. Die Bedeutung der Parameter wurde bereits in Kapitel 3.1.3 beschrieben. Hinweise zur Darstellung von Datenmodellen liefert z.B. [Balzert 99].

Jede Tabelle verfügt über ein Attribut namens „UID“ (unique identifier), was grundsätzlich als Primärschlüssel verwendet wird. Es werden keine zusammengesetzten Primärschlüssel verwendet, da in fast jeder Tabelle Datensätze möglich sind, die sich nur in einem Element unterscheiden. Das bedeutet, dass ein zusammengesetzter Primärschlüssel immer aus einer Kombination aller Attribute bestehen müsste, was im Vergleich zur Verwendung des zusätzlichen UID-Elements einen weitaus größeren Aufwand bedeuten würde.

Ist ein Attribut ein Fremdschlüssel, so besitzt es das Präfix „ID_“. Eine Besonderheit bei der Verwendung von Fremdschlüsseln unter MySQL ist, dass zwar ein tabellenübergreifendes Suchen von Datensätzen mittels Referenzen unterstützt wird, die Integritätsprüfung beim Ändern (UPDATE) oder Löschen (DELETE) jedoch nicht, weshalb die Prüfung im Hauptprogramm ausgeführt wird. (vgl. Abschnitt 4.3.2, Datensatz löschen).

Die Struktur des Datenmodells entspricht den Anforderungen aus Abschnitt 3.1: ein Komplettwerkzeug besteht aus einer Aufnahme und einem Werkzeug, welche ihrerseits aus verschiedenen Komponenten bestehen.

Es ist zu beachten, dass bei Werkzeug und Aufnahme einige Attribute direkt als Wert gespeichert sind, andere hingegen als Verweis auf eine weitere Relation. Ein Attribut wird immer dann direkt gespeichert, wenn es ein Zahlenwert ist oder dessen Inhalt frei formulierbar sein soll (z.B.: Kommentare, Beschreibungen). Eine Relation existiert immer dann, wenn es möglich ist, den einzugebenden Werten eine bestimmte Struktur zu verleihen.

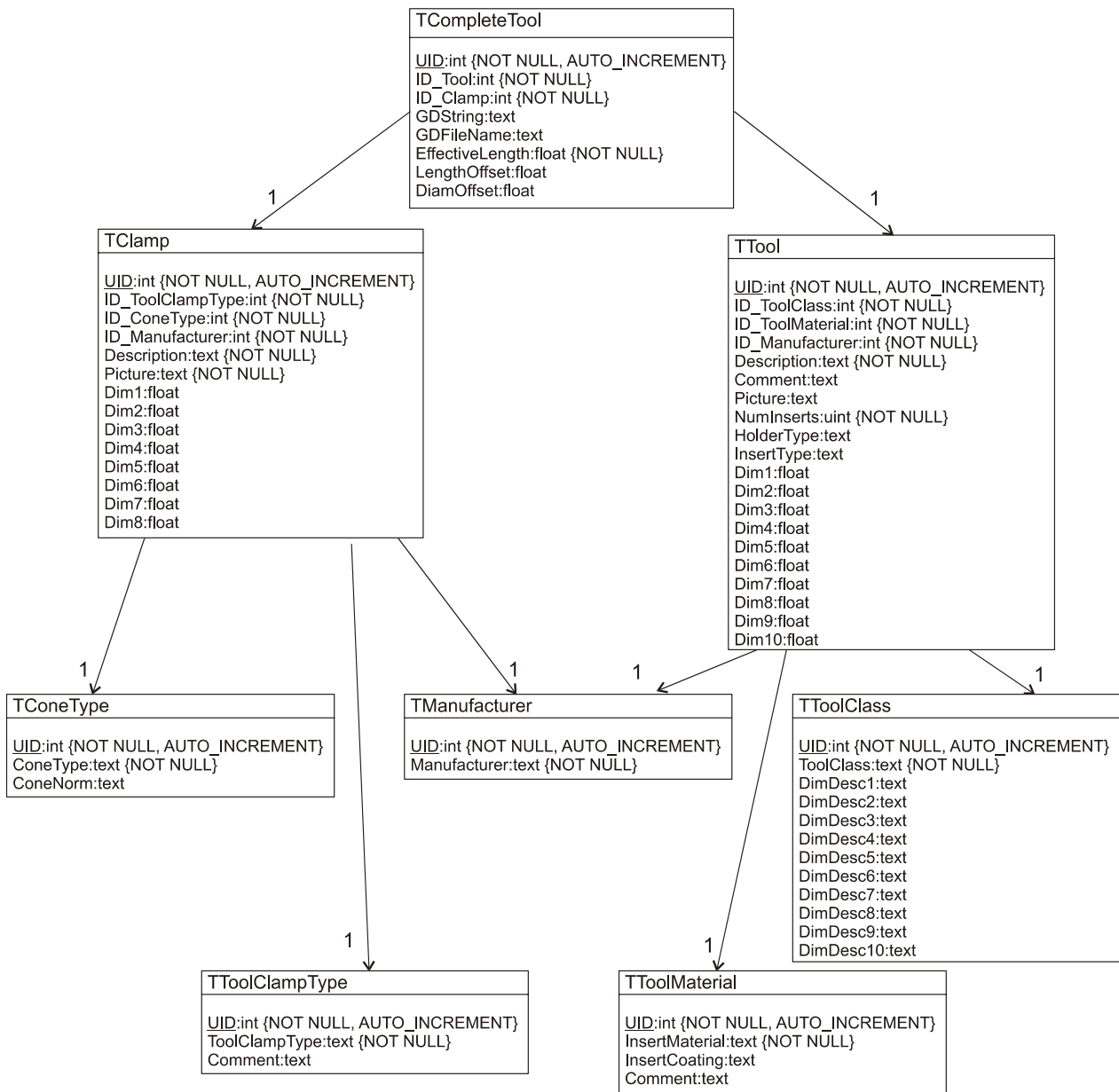


ABBILDUNG 13. Datenmodell

Darstellung der geometrischen Parameter

Die Darstellung der geometrischen Parameter erfolgt für Werkzeuge und Aufnahmen auf unterschiedlichem Weg.

Für jede Werkzeugklasse gibt es unterschiedliche Maße (Schneidendurchmesser, Eckradius, Spitzwinkel, u.s.w.), weshalb es schwierig ist, für jedes mögliche Maß ein eigenes Attribut zu definieren. Stattdessen werden Platzhalter zur Aufnahme beliebiger Maße bereitgestellt.

Nach einer Untersuchung von Werkzeugkatalogen zahlreicher Hersteller¹ hat sich herausgestellt, dass sich die Geometrien aller betrachteten Werkzeuge mit sechs oder weniger Parametern beschreiben lassen. Um zu gewährleisten, dass es keine Probleme beim späteren Hinzufügen von Werkzeugen gibt, unterstützt die Werkzeugverwaltung bis zu zehn Parameter.

Zur Speicherung eines Parameters wird der eigentliche Zahlenwert in die Tabelle 'TTool' eingetragen, in der jeder Eintrag ein gespeichertes Werkzeug darstellt. Die Bedeutung des Wertes steht in der Relation 'TToolClass', in der für jede verfügbare Werkzeugklasse ein Eintrag existiert (siehe Abb. 13). Mit diesem System ist es möglich, jedes beliebige Werkzeug (das sich mit maximal zehn Parametern beschreiben lässt) in die Werkzeugverwaltung aufzunehmen.

Zusätzlich zu den Worterklärungen existiert für jede Werkzeugklasse eine technische Zeichnung, aus der die Bedeutung der Parameter ebenfalls hervorgeht.

Die Eingabe der geometrischen Parameter für Aufnahmen ist einfacher gestaltet, das heißt, es werden keine extra Bezeichnungen für die Maße vergeben. Der Grund dafür ist, dass Tabellenmaße generell nicht mit Namen, sondern nur mit Kürzeln (D1, D2, ...) benannt werden. Da die Semantik der geometrischen Aufnahmeparameter nicht aus den Benennungen hervorgeht, muss zu jedem Aufnahmetyp eine erklärende technische Zeichnung gespeichert werden. Die Maße in der Zeichnung werden mit „D1“, „D2“, u.s.w. benannt. An der Stelle, wo die entsprechenden Werte eingegeben werden, sind die Eingabefelder mit „Dim1“, „Dim2“, u.s.w. beschriftet, um eine Zuordnung zu schaffen. Für häufig verwendete Aufnahmen liegen die Zeichnungen der Werkzeugverwaltung bereits bei.

Wie in Kapitel 3.1.2 beschrieben, werden die Aufnahmen ausschließlich für die optische Darstellung, also nicht für Kollisionsberechnungen während der Simulation verwendet, so dass nur die unbedingt benötigten Maße gespeichert werden. Um die Anzahl der benötigten Platzhalter festzustellen, wurden analog zu den Werkzeugen die Kataloge von Aufnahmen-Herstellern² untersucht. Die Betrachtung ergab, dass zur (vereinfachten) Darstellung aller untersuchten Aufnahmen bis zu vier Parameter benötigt werden. Zur Sicherheit sind in der Werkzeugverwaltung acht geometrische Aufnahmeparameter vorgesehen.

-
1. Arno (www.arno.de), BIAx (www.biax.com), Brütsch/Rügger (www.b-r.ch), Sandvik Coromant (www.coromant.sandvik.com), DEPO (www.depo.de), Fraisa (www.fraisa.ch), Gühring (www.guehring.de), HAM (www.ham-tools.com), Ifanger (www.ifanger.com), ILIX (www.ilix.com), Kaiser (www.heinzkaiser.com), Komet (www.komet.de), Seco (www.secotools.de), TITEX (www.titex.com), Plansee Tizit (www.plansee.com), Widia (www.widia.com)
 2. Brütsch/Rügger (www.b-r.ch), Sandvik Coromant (www.coromant.sandvik.com), D'Andrea (www.dandrea.com), Diebold (www.diebold-werkzeuge.de), GEWEFA (www.gewefa.de), Kaiser (www.heinzkaiser.com), Kieninger (www.kieninger.de), Lyndex Corp (www.lyndex.com), Seco (www.secotools.de), WIDIA (www.widia.com)
-

4.2.2 Klassendiagramm der Hauptanwendung

Die Werkzeugverwaltung ist eine dialogfeldbasierte Anwendung, was bedeutet, dass kein Dokument im Hintergrund existiert, welches die Daten des Programms hält. Alle Fenster sind dementsprechend Dialoge, die jeweils von der MFC-Klasse 'CDialog' abgeleitet sind. Die wichtigsten vier Dialoge sind das Hauptfenster (CDlgMain) und die Masken zur Auswahl von Werkzeugen, Aufnahmen und Komplettwerkzeugen (CDlgChooseTool, CDlgChooseClamp und CDlgCompleteTool).

Im Abbildung 14 werden die Abhängigkeiten der verschiedenen Klassen verdeutlicht.

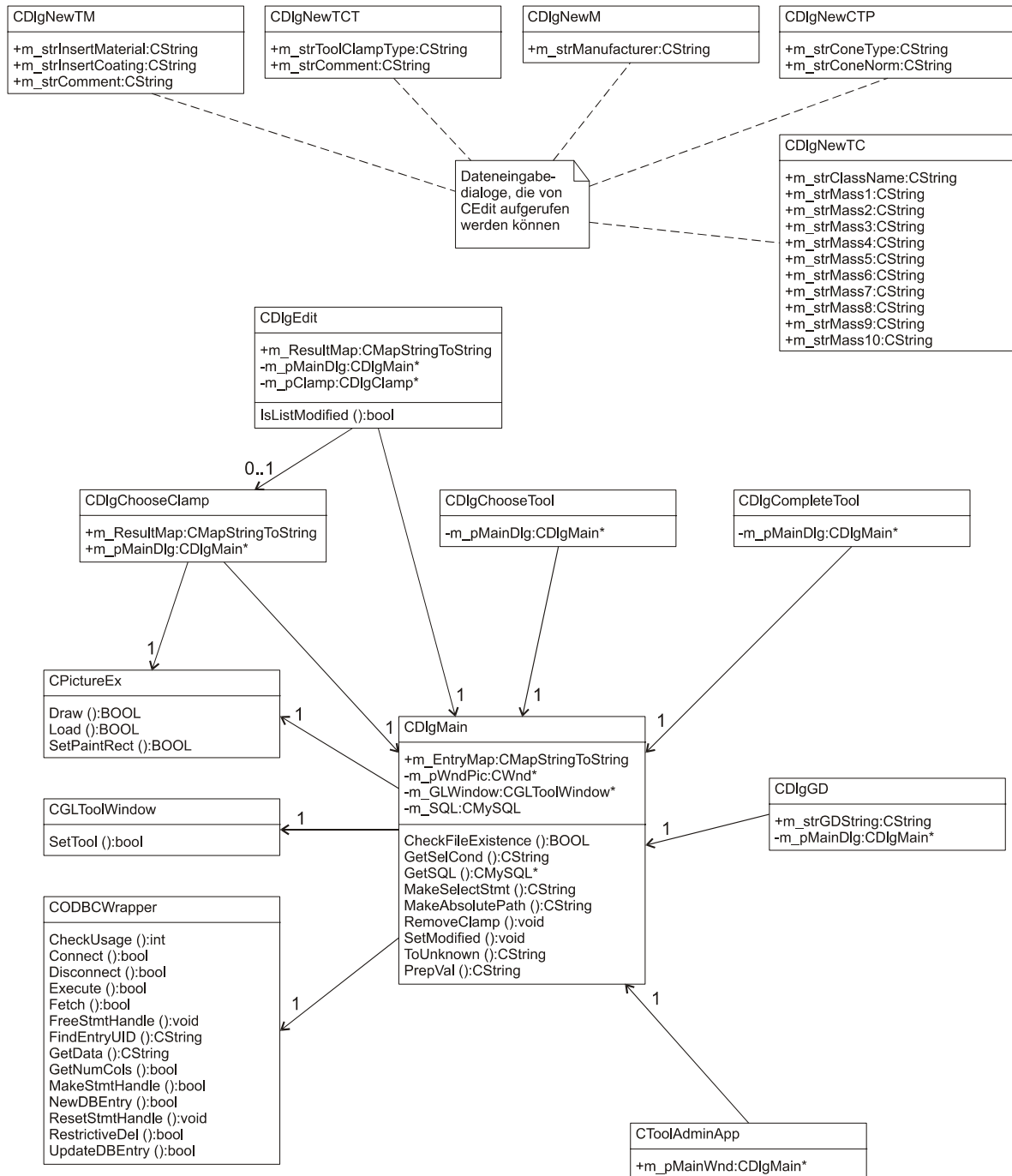


ABBILDUNG 14. Klassendiagramm der Hauptanwendung

4.3 Erläuterungen zum Quellcode

4.3.1 Dialogstruktur

Grundstruktur der Anwendung

Die Entwicklungsumgebung 'MS Visual Studio' stellt einen MFC-Anwendungsassistenten zur Verfügung, mit dem die Grundstruktur der Anwendung erstellt wurde. Eine gute Referenz zur Benutzung des Visual Studios und der MFC bietet [Prosis 99]. Zur Grundstruktur gehören die Klassen 'CToolAdminApp' und 'CDlgMain'. Abbildung 15 zeigt den Hauptdialog mit einem ausgewählten Komplettwerkzeug.

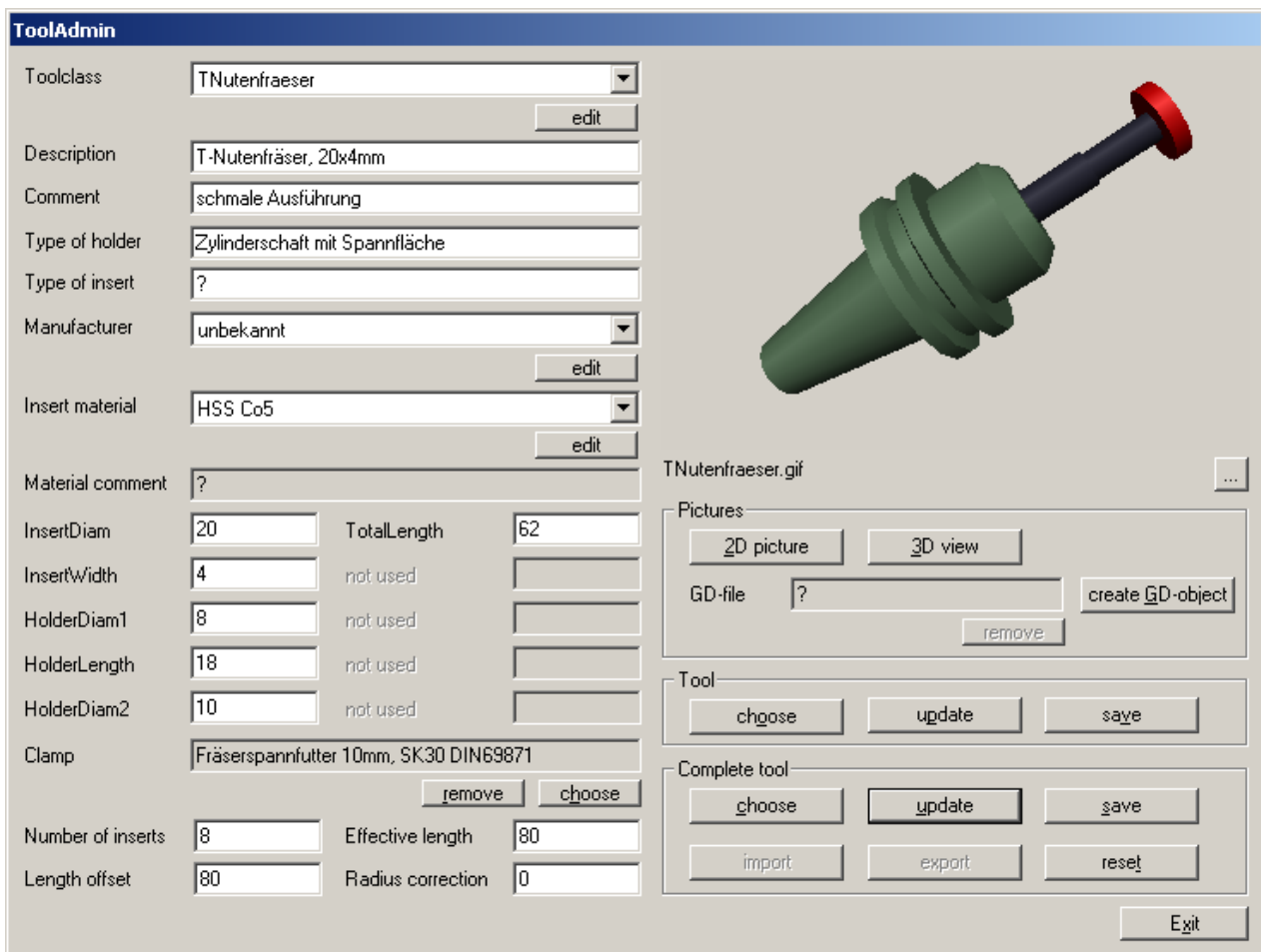


ABBILDUNG 15. Hauptdialog der Werkzeugverwaltung

'CToolAdminApp' ist von 'CWinApp' abgeleitet und überschreibt lediglich die Methode 'InitInstance', die hauptsächlich dafür sorgt, dass das Hauptfenster zu Programmstart angezeigt wird. Von 'CToolAdminApp' wird eine einzige globale Instanz angelegt, welche die Anwendung selber repräsentiert.

'CDlgMain' ist die Klasse für das Hauptfenster der Werkzeugverwaltung, das Ausgangspunkt für alle Anwendungsfälle ist. Die beiden Member-Objekte 'm_SQL' und 'm_EntryMap' sind für die Anwendung von besonderer Bedeutung und werden deshalb genauer erklärt.

Das Objekt 'm_SQL' ist eine Instanz der Klasse 'CODBCWrapper' - die einzige im gesamten Programm. Alle anderen Dialoge, von denen aus SQL-Befehle ausgeführt werden sollen, müssen sich einen Zeiger über die Zugriffsfunktion 'GetSQL' besorgen. Der Vorteil hierbei ist, dass nur einmal zu Programmstart eine Datenbankverbindung hergestellt werden muss (vgl. Abs. 4.3.2). Verfügte jeder Dialog über ein eigenes SQL-Objekt, müsste bei fast jedem Dialogaufruf eine Verbindung zur Datenbank erstellt werden, um die jeweiligen SQL-Objekte zu initialisieren.

Ebenfalls von zentraler Bedeutung ist das Objekt 'm_EntryMap', was vom MFC-Typ 'CMapStringToString' ist. Eine Map ist eine Tabelle, in der Elemente paarweise gespeichert werden, wobei der erste Eintrag ein Schlüssel ist, über den der Zweite gefunden werden kann. In der Werkzeugverwaltung speichert die Map alle Informationen über das aktuell ausgewählte Werkzeug, indem für jedes mögliche Attribut ein Eintrag in der Map erzeugt wird. Das Schlüsselement ist jeweils der Name einer Tabellenspalte mit vorangehendem Bezeichner der Relation selber, z.B.: „TTool.Manufacturer“. Der assoziierte Eintrag ist der zugehörige Wert, z.B.: „Coromant“.

Wird ein Dialog zum Editieren des aktuellen Werkzeugs aufgerufen, werden ggf. dessen Felder mit dem Inhalt der Map initialisiert. Beim Beenden des Dialogs werden die Änderungen in der Map gespeichert. Damit ist 'm_EntryMap' das zentrale Objekt zur Darstellung des aktuellen Werkzeugs.

Resourcendateien:

Um bei Namensänderungen von Tabellen oder Spalten in der Datenbank nicht den gesamten Quellcode editieren zu müssen, sind in dem Header 'DBResource.h' Aliasnamen für Spalten und Tabellen definiert, die durchgängig im Quellcode verwendet werden.

Sämtliche Dialogtexte - also Steuerelementbeschriftungen, Hinweis- und Fehlermeldungen - sind in der Resourcendatei ToolAdmin.rc gespeichert. Sollte die Software zu einem späteren Zeitpunkt in eine andere Sprache übersetzt werden, können alle Texte zentral geändert werden.

Dialoge zur Verwaltung von Werkzeugen und Aufnahmen

Die wichtigsten Fenster der Werkzeugverwaltung sind, neben dem Hauptfenster, die Dialoge zur Auswahl von Werkzeugen und Komplettwerkzeugen, sowie der Dialog zur Verwaltung von Aufnahmen. Sie werden alle aus dem Hauptfenster gestartet und sind im folgenden einzeln beschrieben.

Werkzeugauswahl (CDlgChooseTool)

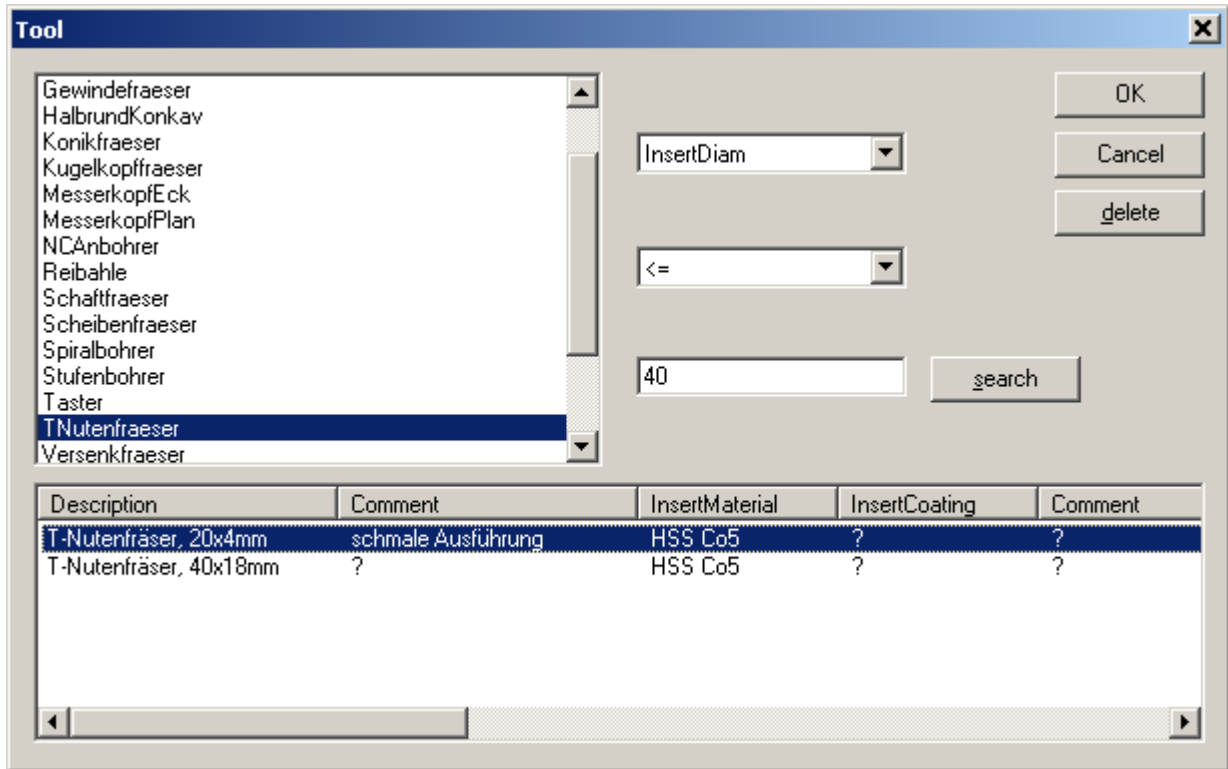


ABBILDUNG 16. Werkzeugauswahldialog

Der in Abbildung 16 dargestellte Dialog ermöglicht die Selektion einzelner Werkzeuge aus der Datenbank, wobei die Auswahl eines Werkzeugs in drei Stufen geschieht: als erstes wird aus einer Liste die gewünschte Werkzeugklasse bestimmt (z.B.: „Schaftfräser“), woraufhin alle verfügbaren Werkzeuge in einer Tabelle angezeigt werden. Ist die Zahl der Einträge zu groß, kann als zweites ein Filter gesetzt werden, der die Auswahl eingrenzt (z.B.: „Schneidendurchmesser < 8“). Zuletzt wird das gewünschte Werkzeug ausgewählt, wodurch alle Attribute in die 'EntryMap' (siehe vorherige Seite) des Hauptfensters übergeben werden.

Auswahl eines Komplettwerkzeugs (CDlgCompleteTool)

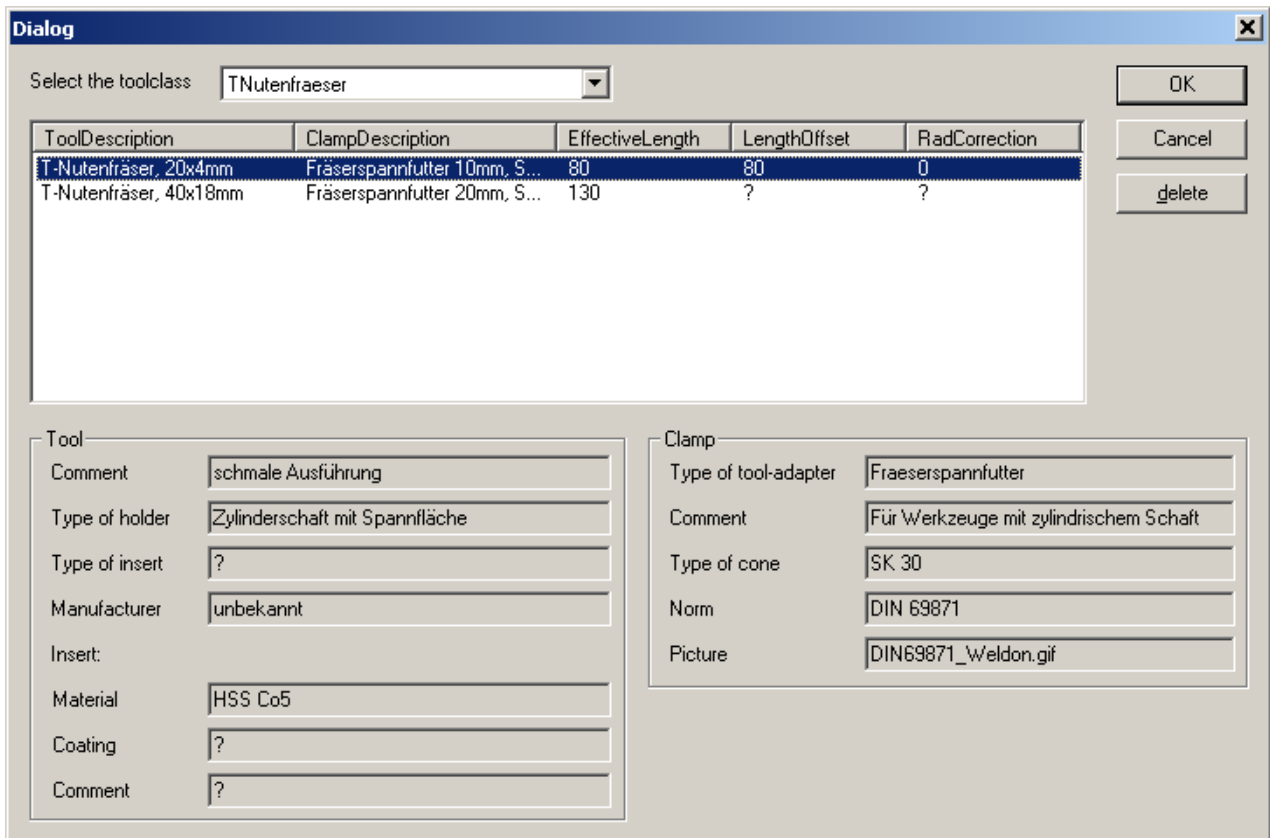


ABBILDUNG 17. Dialog zur Auswahl von Komplettwerkzeugen

Wie in Kapitel 3.2.1 beschrieben, wird dieser Dialog (Abbildung 17) sehr häufig aufgerufen, weshalb die Auswahlprozedur möglichst simpel gestaltet ist. Die Klasse des gewünschten Werkzeugs wird in einer Liste ausgewählt, woraufhin alle hierzu gefundenen Komplettwerkzeuge in einer Tabelle dargestellt werden. Nach Auswahl eines Werkzeugs werden die wichtigsten Informationen, nach Gruppen geordnet, im Auswahldialog dargestellt. Nachdem der Dialog mit 'OK' beendet wurde, werden alle Daten zum Komplettwerkzeug in die 'EntryMap' des Hauptfensters übergeben.

Verwaltung der Aufnahmen (CDlgChooseClamp)

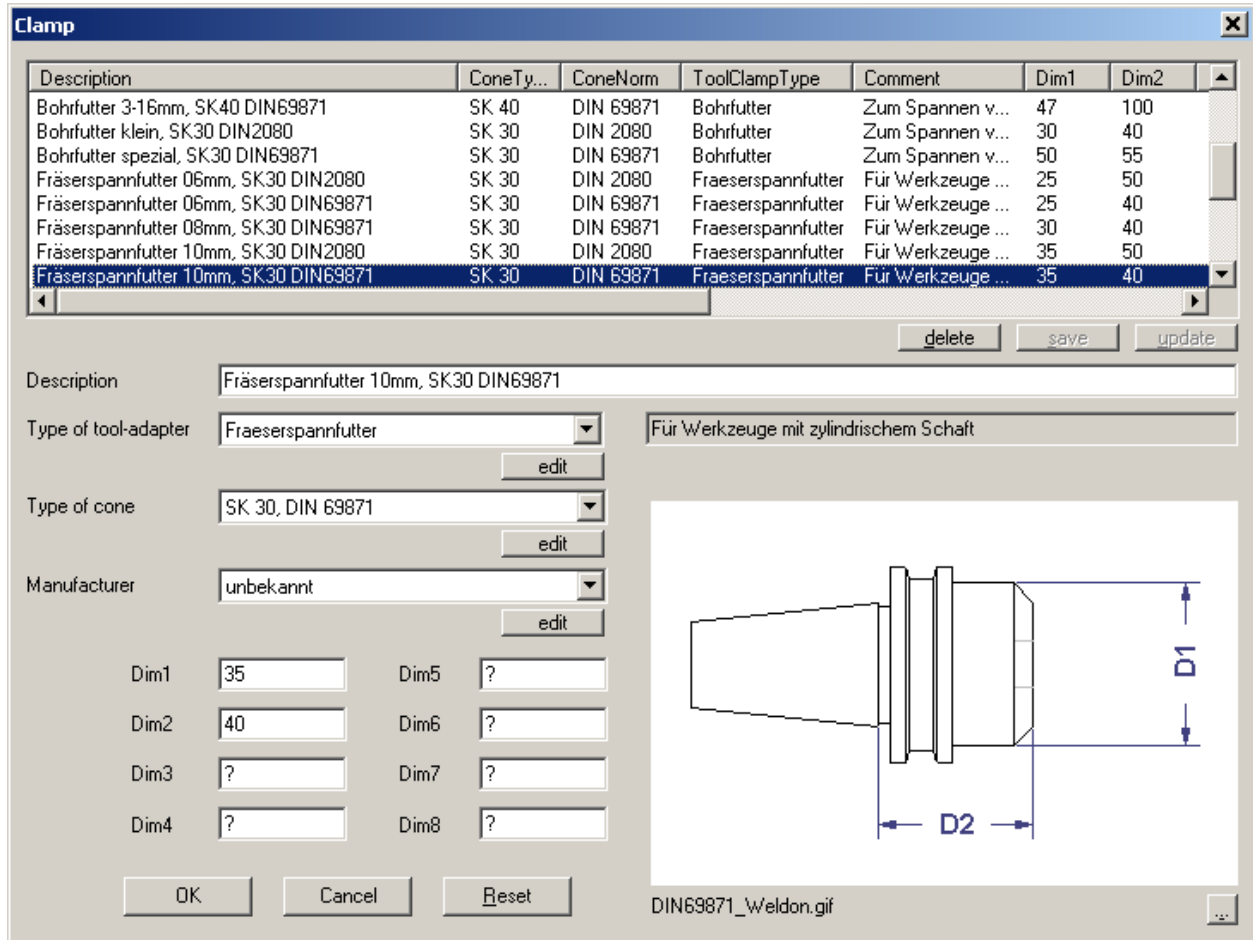


ABBILDUNG 18. Dialog zur Verwaltung von Aufnahmen

Der in Abbildung 18 gezeigte Dialog wird hauptsächlich zum Auswählen von Aufnahmen aufgerufen. Er ermöglicht jedoch auch deren Erstellung und Modifikation (bei den komplexeren Werkzeugen werden diese Aufgaben vom Hauptdialog übernommen). Da in diesem Dialog Änderungen vorgenommen werden können, die aber nicht unbedingt übernommen werden müssen (der Dialog kann mit 'Cancel' beendet werden), verfügt er über eine eigene Map zum Speichern von Aufnahmeparametern. Wenn der Dialog mit 'OK' beendet wird, werden dessen Daten in die 'EntryMap' des Hauptfensters kopiert.

Dialoge zum Verwalten von Hilfstypen

Als Hilfstypen werden alle Parameter bezeichnet, die in einer eigenen Tabelle verwaltet und von Werkzeugen oder Aufnahmen referenziert werden - also Schneidstoffe, Werkzeugklassen, Hersteller, Werkzeugadapter und Kegeltypen (vgl. Abb.13).

Die Verwaltung von Hilfstypen kann aus dem Hauptdialog oder dem Werkzeugauswahlfenster aufgerufen werden, indem eine Schaltfläche 'edit' angeklickt wird. Daraufhin wird ein Dialog (siehe

Abbildung 19) der Klasse 'CDlgEdit' geöffnet, der in einer Tabelle alle Einträge zu dem entsprechenden Hilfstyp anzeigt und eine Reihe von Optionen zur Verfügung stellt: 'new', 'edit' und 'delete'.



ABBILDUNG 19. Editierdialog (für Werkzeugmaterialien)

Mit 'delete' wird der aktuell ausgewählte Eintrag gelöscht (vgl. Abs. 4.3.2, Datensatz löschen). Soll ein Datensatz neu erstellt werden, wird ein Eingabedialog (Abbildung 20) geöffnet, wobei es für jeden Hilfstyp eine eigene Maske gibt. Zum Ändern eines Eintrags wird die gleiche Maske benutzt und mit den Werten des aktuell markierten Datensatzes initialisiert.

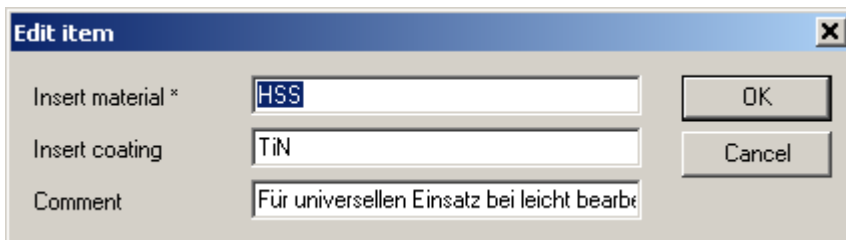


ABBILDUNG 20. Eingabemaske für Werkzeugmaterialien

Analog zum Dialog für Aufnahmen müssen die Einstellungen nicht übernommen werden, weshalb der Editierdialog ebenfalls über eine Map zum Zwischenspeichern der Parameter verfügt.

Anzeige von technischen Zeichnungen und 3D-Objekten

Für das Darstellen der 2D-Zeichnungen ist die Klasse 'CPictureEx' zuständig, die von [Bykov 03] stammt. Der Vorteil dieser Klasse liegt darin, dass man verschiedene gängige Grafikformate anzeigen lassen kann, während die MFC-Standardklassen lediglich die Darstellung von Bitmaps (bmp) unterstützen. In Abschnitt 4.5 finden sich detaillierte Angaben zur Gestaltung der Zeichnungen.

Zur Darstellung der 3D-Werkzeuge wird die Klasse 'CGLToolWindow' verwendet, die als 'dynamic link library' (dll) eingebunden und in diversen Anwendungen der STA GmbH eingesetzt wird. In der Werkzeugverwaltung wird das Fenster mit den selben Koordinaten erzeugt, die auch für die 2D-Darstellung benutzt werden. In Abhängigkeit des angezeigten Objekts ist immer eines der beiden Fenster sichtbar während das Andere versteckt ist (in Abb. 15 ist ein T-Nutenfräser als 3D-Objekt dargestellt).

Alles, was für die Darstellung eines 3D-Objekts nötig ist, ist der Aufruf der Methode 'SetTool', deren einziger Parameter ein kompletter GD-String ist.

Erzeugen von GD-Objekten

Ist ein Komplettwerkzeug ausgewählt, kann im Hauptfenster die Schaltfläche zum Erstellen einer GD-Datei angeklickt werden. Es erscheint ein Dialog des Typs 'CDlgGD', der zur Eingabe eines Dateinamens auffordert. Wird kein Pfad angegeben, so erstellt die Werkzeugverwaltung die Datei in einem Standardverzeichnis, relativ zur Hauptanwendung.

```
\ToolAdmin\GD\..
```

Optional kann die Erstellung einer GD-Datei entfallen, so dass der GD-String nur in der Datenbank gespeichert wird, was die Anzeige des 3D-Werkzeugs im Hauptfenster ermöglicht.

Die genauen Abläufe bei der Erstellung von GD-Objekten werden in Kapitel 4.4 erläutert.

4.3.2 Die Klasse CODBCWrapper

Wie der Name schon sagt, baut diese Klasse eine Hülle um Funktionen der ODBC API, um diese einfacher im Quellcode benutzen zu können. Da die Methoden sehr häufig in der Werkzeugverwaltung benutzt werden, sind sie in diesem Abschnitt ausführlich beschrieben.

Auf- und Abbau der Datenbankverbindung

Der Aufbau einer Datenbankverbindung geschieht in mehreren Schritten. Als erstes wird ein 'Environment-Handle' erzeugt, der Zugang zu globalen Informationen über die Datenbankverbindung liefert und eine Verbindung zum ODBC-Treiber schafft. Als nächstes wird über den Environment-Handle die verwendete ODBC-Version festgelegt (die aktuelle Version ist 3.51). Um die Verbindung(en) zu einem Datenbankserver zu verwalten, muss ein 'Connection-Handle' erzeugt werden, der über Informationen über die Gültigkeit von 'Statement-Handles' (siehe nächste Seite), Transaktionen und Ähnlichem bereithält.

Nun kann die eigentliche Verbindung zur Datenbank initialisiert werden. Die wesentlichen Parame-

ter zum Verbindungsaufbau sind der 'Data Source Name' (DSN) über den eine Datenbank angesprochen wird und ein optionaler Benutzername mit dem zugehörigen Kennwort.

Die Methode 'Connect' der Klasse CODBCWrapper, die einmalig im Konstruktor des Hauptfensters aufgerufen wird, erledigt alle diese Aufgaben. Zum Schließen der Verbindung und zum Freigeben der Handles wird entsprechend im Destruktor eine Methode 'Disconnect' aufgerufen.

Verwalten von Statement-Handles

Bevor ein SQL-Befehl ausgeführt werden kann, muss ein 'Statement-Handle' erzeugt werden, der Informationen wie Fehlermeldungen, Abarbeitungsstatus einer Anfrage, Cursor (Zeiger auf ein Element einer Ergebnismenge, die durch einen Suchbefehl entstanden ist), u.Ä. verwaltet.

Des Weiteren können über einen Statement-Handle Attribute für SQL-Befehle gesetzt werden, zum Beispiel eine maximal erlaubte Anzahl von Ergebnissen für eine Suchabfrage.

Die Wrapperklasse stellt drei Funktionen zur Verwaltung von Statement-Handles zur Verfügung:

- **MakeStmntHandle:** erzeugt einen neuen Handle und setzt das Attribut 'SQL_SCROLLABLE', wodurch in der Ergebnismenge eines Suchbefehls beliebig navigiert werden kann.
- **ResetStmntHandle:** bereitet den Handle für einen neuen Befehl vor, d.h. Fehlermeldungen, Cursor, Statusflags, u.s.w werden gelöscht.
- **FreeStmntHandle:** löscht einen Statement-Handle.

Nachdem ein Statement-Handle angelegt wurde, können SQL-Befehle an die Datenbank gesendet werden.

Auswählen von Datensätzen (SELECT)

Zum Ausführen eines SQL-Befehls wird die Methode 'Execute' ausgeführt, die als Parameter einen Statement-Handle und den SQL-Befehl als Zeichenkette übernimmt.

Handelt es sich um einen Suchbefehl, so wird nach erfolgreicher Ausführung eine Ergebnismenge mit allen gefundenen Einträgen zurückgeliefert. Hieraus können mit der Methode 'Fetch' einzelne Datensätze ausgewählt werden. Da die Statement-Handles grundsätzlich über das Attribut 'SQL_SCROLLABLE' (siehe oben) verfügen, können die Datensätze sowohl relativ als auch absolut adressiert werden, d.h., die Einträge können schrittweise durchlaufen oder direkt über Indizes ausgewählt werden. Die Art der Adressierung wird durch einen Parameter der Funktion Fetch festgelegt ([MySQL 03] liefert eine ausführliche Befehlsreferenz im Internet).

Ist ein Datensatz ausgewählt, kann auf dessen einzelne Elemente über die Methode 'GetData' zugegriffen werden, deren Parameter der Statement-Handle und ein Index der gewünschten Spalte sind.

Wird die Zahl der gefundenen Einträge benötigt, kann sie über die Methode 'GetNumCols' ermittelt werden.

Neue Datensätze erstellen (INSERT)

Zum Einfügen von Datensätzen dient die Methode 'NewDBEntry'. Sie erwartet als Parameter einen Tabellennamen und eine Liste der einzufügenden Werte. Intern wird daraus ein SQL-Befehl (INSERT INTO <Tabelle> VALUES <Daten>) geformt, der mittels der Funktion 'Execute' (siehe vorherige Seite) ausgeführt wird. 'NewDBEntry' gibt einen booleschen Wert zurück, der nach erfolgreichem Einfügen wahr wird.

Datensatz ändern (UPDATE)

Das Ändern von Datensätzen geschieht über die Methode 'UpdateDBEntry'. Der zugehörige SQL-Befehl lautet: „UPDATE <Tabelle> VALUES <Daten> WHERE <Bedingungen>“. Dementsprechend übernimmt die Methode einen Tabellennamen, eine Liste der zu ändernden Daten und eine Liste von Bedingungen, die festlegt, in welchen Datensätzen die Änderungen durchgeführt werden sollen. Ein boolescher Rückgabewert zeigt an, ob das Ändern erfolgreich war.

Vor dem Aufruf der Methode 'UpdateDBEntry' wird immer die Funktion 'CheckUsage' ausgeführt. Diese prüft, ob einer der zu verändernden Datensätze von einem Anderen referenziert wird. Ist das der Fall, erscheint eine Hinweismeldung, und der Benutzer hat die Möglichkeit den Vorgang abbrechen.

Inkonsistenzen in der Datenbank können beim Ändern und Hinzufügen nicht auftreten (indem ein ungültiger Fremdschlüssel gesetzt wird), da seitens der Werkzeugverwaltung keine ungültigen Referenzen gesetzt werden können.

Datensatz löschen (DELETE)

Eine Besonderheit von MySQL ist, dass kein restriktives Löschen unterstützt wird. Das Entfernen eines Datensatzes, der noch von einem Anderen referenziert wird, wird also ohne eine Fehlermeldung durchgeführt. Um eine Inkonsistenz der Datenbank zu vermeiden, muss also vor jedem Löschen geprüft werden, ob der Vorgang überhaupt erlaubt ist. Die Methode 'RestrictiveDel' kommt genau dieser Aufgabe nach. Ihr wird der Name der Tabelle, in der gelöscht werden soll und die UID des zu löschenden Eintrags übergeben. 'RestrictiveDel' prüft nun, ob diese UID in einer anderen Tabelle als Fremdschlüssel verwendet wird. Ist dem so, wird das Löschen nicht durchgeführt und eine Fehlermeldung ausgegeben. Andernfalls wird der Datensatz entfernt.

4.4 TCL-Skripte

4.4.1 Verwendung der Tcl-Skripte

Wie in Kapitel 3.2.2 beschrieben wurde, existiert für jede Werkzeug- und Aufnahmeklasse ein eigenes Skript zur Erzeugung der Geometriedaten-Dateien. Um Änderungen in den Skripten einfach zu gestalten und eine größtmögliche Modularität zu erreichen, sind die Skripte in vielen einzelnen Dateien und nicht in einer Großen abgelegt. Es ist nicht unbedingt erforderlich, Skripte bereitzustellen, da sie ausschließlich der Erzeugung von 3D-Objekten dienen. Möchte man jedoch GD-Objekte aus Datenbankeinträgen erzeugen, so sind die entsprechenden Tcl-Dateien unabdingbar.

Falls in der Werkzeugverwaltung neue Werkzeug- oder Aufnahmetypen erzeugt wurden und zugehörige Skripte erstellt werden sollen, so ist darauf zu achten, sie genau wie die Klassen zu benennen, denen sie zugeordnet sind. Hierdurch ist es beim Erzeugen von GD-Objekten möglich, automatisch die benötigten Skripte zu finden. Eine Alternative hierzu wäre eine Tabelle, welche die einzelnen Dateinamen mit den zugehörigen Klassen assoziiert, was jedoch einen höheren Verwaltungsaufwand mit sich brächte. Ein Skript für die Werkzeugklasse 'Fasfraeser' heisst demzufolge 'Fasfraeser.tcl'. Für eine Aufnahme werden zwei Dateien benötigt: eine für den Kegel und eine für den Werkzeugadapter. Ein Bohrfutter mit einem SK 30 Kegel nach DIN 2080 wäre zum Beispiel durch die beiden Dateien 'Bohrfutter.tcl' und 'SK30DIN2080.tcl' beschrieben. Der Vorteil dieser Zweiteilung liegt darin, dass Kegel und Werkzeugadapter beliebig kombiniert werden können, ohne dass redundante Daten gehalten werden müssen.

Alle Tcl-Dateien liegen relativ zur Hauptanwendung und sind wie folgt angeordnet:

```
\ToolAdmin\TCL\..
    \Tool\Werkzeugklasse.tcl
    \ToolAdapter\Adaptername.tcl
    \ClampCone\Kegeltyp.tcl
    \TclMain.tcl
```

4.4.2 Erläuterung der einzelnen Skripte

TclMain.tcl

Die Datei 'TclMain.tcl' enthält keine Anweisungen zum Erstellen einer Geometrie, sondern bildet eine Schnittstelle zwischen den Geometrieskripten und der Werkzeugverwaltung. Diese Datei bleibt daher stets unverändert, unabhängig davon, ob Geometrieskripte modifiziert, gelöscht oder hinzugefügt werden.

Sie stellt drei Prozeduren bereit, die von der Werkzeugverwaltung der Reihe nach aufgerufen werden müssen, um einen GD-String zu erzeugen. Die ersten beiden Prozeduren ('SetPath' und 'Init') die-

nen der Initialisierung, während die Dritte ('CreateGDString') die Erzeugung der Geometrie übernimmt.

SetPath (char* pszBaseDir)

Diese Funktion sorgt dafür, dass ein absoluter Pfad zum Verzeichnis '\TCL' (siehe Abschnitt 4.4.1) in einer globalen Variable im Tcl-Skript gespeichert wird, wobei 'pszBaseDir' das Arbeitsverzeichnis der Hauptanwendung enthalten muss. Zum Finden der Geometrieskripte wird relativ zum Verzeichnis '\TCL' gesucht, was bedeutet, dass die in Abschnitt 4.4.1 festgelegte Verzeichnisstruktur eingehalten werden muss.

Init (char* pszParams)

Hier werden alle Variablen des Skripts mit den Werten des aktuell ausgewählten Werkzeugs initialisiert. Bei *pszParams* handelt es sich um eine Zeichenkette, die - durch Kommata getrennt - alle zu übergebenden Parameter beinhaltet:

- Werkzeugmaß 1 bis 10 (*tDim1 bis tDim10*)
- Aufnahmemmaß 1 bis 8 (*cDim1 bis cDim8*)
- Spannlänge (*effLength*)
- Längenoffset (*lOffset*)
- Radiuskorrektur (*dOffset*)
- Werkzeugklasse (*toolClass*)
- Werkzeugaufnahme (*TCT (->ToolClampType)*)
- Kegeltyp (*coneType*)

Diese Parameter werden immer übergeben, unabhängig davon, ob die Werte in der Werkzeugverwaltung gesetzt wurden oder nicht. Es wird allerdings in der Funktion geprüft, ob Längenoffset, Werkzeugklasse, Werkzeugaufnahme und Kegeltyp gesetzt sind. Ist das nicht der Fall, wird eine Fehlermeldung ausgegeben und die Geometrie nicht erzeugt. Sind Längenoffset und Radiuskorrektur nicht gesetzt, werden sie mit Null initialisiert. Die Validierung der Werkzeug- und Aufnahme-maße erfolgt erst in den Geometrieskripten.

CreateGDString ()

Diese Prozedur erzeugt einen GD-String, wie er in Abschnitt 2.2 dargestellt ist, sie darf allerdings erst nach den beiden Initialisierungsroutinen aufgerufen werden. Abbildung 21 zeigt eine 2D-Kontur für ein Komplettwerkzeug und das zugehörige Koordinatensystem.

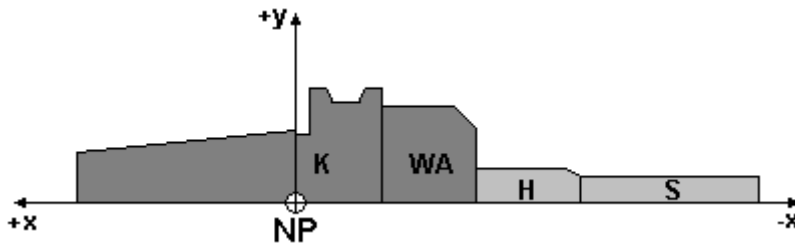


ABBILDUNG 21. 2D-Kontur eines Komplettwerkzeugs (NP=Werkzeugnullpunkt; K=Kegel der Aufnahme; WA=Werkzeugadapter der Aufnahme; H=Halter; S=Schneide)

Im folgenden wird die Verwendung der in 'Init ()' definierten Variablen erläutert.

Die Parameter Längenoffset und Radiuskorrektur werden für die Attribute 'nosePoint' und 'noseRadius' des Knotens 'Tool' verwendet (siehe Kapitel 3.1.3, Längenoffset und Radiuskorrektur).

```
#STA GD V2.0
Tool {
  nosePoint 0 0 -($lOffset)
  noseRadius $dOffset
  ...
}
```

Wie in Abschnitt 2.2 beschrieben, besitzt der Knoten 'Tool' die drei untergeordneten Knoten 'insert', 'holder' und 'clamp', die entsprechend für Schneide, Halter und Aufnahme stehen. Für jeden dieser drei Knoten kann eine Translation und Rotation definiert werden, die sich auf alle untergeordneten Objekte auswirkt. Wie in Abbildung 21 zu sehen ist, liegt der Nullpunkt des Koordinatensystems in der Aufnahme, weshalb diese nicht verschoben wird. Schneide und Halter hingegen werden um die Ausspannlänge des Werkzeugs verschoben, was den Vorteil hat, dass die Schneiden- und Halterkonturen im Tcl-Skript so angegeben werden können, als wäre der Werkzeugnullpunkt gleichzeitig die Werkzeugspitze.

```
insert Transform {
  children Transform {
    translation 0 0 -($effLength)
    ...
  }
holder Transform {
  children Transform {
    translation 0 0 -($effLength)
    ...
  }
}
```

Die Parameter 'Werkzeugklasse', 'Werkzeugaufnahme' und 'Kegeltyp' dienen dazu, die richtigen Geometrieskripte automatisch aufrufen zu können (siehe Abschnitt 4.4.1).

Mit den bis hier beschriebenen Parametern ist bereits das komplette allgemeine Rahmengerüst des GD-Strings erstellt, es fehlen noch die speziellen 2D-Konturen für die Rotationskörper. Wenn die benötigten Geometrieskripte aufgerufen werden, um die Konturen zu erstellen, werden die geometrischen Parameter mit übergeben. Dabei ist zu beachten, dass immer alle verfügbaren zehn, bzw. acht Parameter übergeben werden und nicht nur jene, die von der jeweiligen Klasse gerade benötigt werden. Das liegt daran, dass 'TclMain.tcl' keine genauen Informationen über die zu erstellenden Objekte besitzt; diese sind ausschließlich in den Geometrieskripten gegeben.

Skripte für Schneide und Halter

Da bei einem Fräswerkzeug die Schneide und der Halter normalerweise fest verbunden sind (eine Ausnahme bilden Fräser mit Schneidplatten, die von der Werkzeugverwaltung jedoch wie einteilige Fräser behandelt werden), stehen die Skripte für beide Komponenten in einer gemeinsamen Datei namens *<Werkzeugklassenname.tcl>*.

```
1. proc genInsert {ID IL unused1 unused2 unused3 unused4 unused5 unused6 unused7
2.             unused8} {
3.     if {![isSet $ID]} { error "Tool->InsertDiam is not set!" }
4.     if {![isSet $IL]} { error "Tool->InsertLength is not set!" }
5.     if {$ID<=0} { error "InsertDiam must be > 0" }
6.     if {$IL<=0} { error "InsertLength must be > 0" }
7.
8.     set IR [expr $ID/2.0]
9.     set cont [genContour "0 0, \
10.                    0 $IR, \
11.                    $IL $IR, \
12.                    $IL 0"]
13.     return $cont
14. }
```

Das Beispiel zeigt die Prozedur 'genInsert' aus der Datei 'Bohrnutenfraeser.tcl'. Die Prozedur 'genHolder', die ebenfalls in der Datei steht, wurde nicht mit abgebildet, da der Inhalt analog zum Beispiel ist. Die Zahlen am Anfang der Zeilen sind nur zur besseren Übersicht eingefügt worden.

Bei der Parameterliste ist zu erkennen, dass alle zehn Werkzeuggeometrien übergeben werden, allerdings nur die ersten Beiden verwendet werden. Die restlichen acht Werte werden einfach mit 'unusedX' bezeichnet und können im weiteren ignoriert werden.

Die Zeilen 3 bis 6 dienen der Wertprüfung. Ist einer der beiden geprüften Werte kleiner als Null oder überhaupt nicht gesetzt, so wird eine Fehlermeldung ausgegeben und die Erstellung des GD-

Strings abgebrochen.

In Zeile 8 wird eine Hilfsvariable 'IR' (->InsertRadius) definiert, um die Konturerstellung übersichtlicher zu gestalten.

Das Erstellen der Schneidenkontur erfolgt in den Zeilen 9 bis 12, indem eine Variable 'cont' erstellt und mit einer Liste von Wegelementen gefüllt wird. Die Streckenabschnitte werden jedoch nicht direkt abgespeichert, sondern erst mittels der Funktion 'genContour' modifiziert. Der Grund dafür ist, dass sich eine Kontur nicht nur aus Geradenelementen zusammensetzt (wie in diesem einfachen Beispiel), sondern ein Streckenzug aus beliebig zusammengesetzten Kreis- Ellipsen- und Geradenelementen sein kann. Ein Streckenabschnitt wird normalerweise als 6-Tupel angegeben; die genaue Syntax ist in [STA 03] beschrieben. Die Funktion 'genContour' ergänzt die fehlenden Werte für jedes angegebene Streckenelement, sodass der Eintrag „0 \$IR“ zu „0 0 \$IR 0 0 0“ wird.

Nachdem die Kontur vollständig erstellt ist, wird sie von der Prozedur zurückgegeben.

Eine Besonderheit existiert bei den Skripten für die Aufnahmekegel. Da die Geometrie eines Kegels vollständig genormt ist, müssen hier keine Parameter übergeben werden und die Plausibilitätsprüfung kann ebenfalls entfallen. Das folgende Beispiel gibt den Inhalt der Datei „SK50DIN2080“ wieder:

```
proc genClampCone {} {
  global fixtureLength

  set fixtureLength 15.2
  set FL $fixtureLength
  set cont [genContour "-$FL          48.75, \
                      -[expr $FL-12] 48.75, \
                      -[expr $FL-12] 34.925, \
                      0              34.925, \
                      101.8          19.8, \
                      126.8          19.8, \
                      126.8          0"]

  return $cont
}
```

Die globale Variable 'fixtureLength' gibt den Abstand vom Werkzeugnullpunkt bis zum werkzeugseitigen Ende des Aufnahmekegels an. Sie wird für den Werkzeugadapter benötigt, der genau um diesen Abstand in Richtung -x (siehe Abb. 21) verschoben werden muss, um nicht im Kegel gezeichnet zu werden. Da die Variable während der Kegelerstellung gesetzt wird, muss die Erzeugung des Werkzeugadapters zwangsläufig danach erfolgen.

4.5 Zeichnungen zu Werkzeugen und Aufnahmen

Für die Erstellung der 2D-Zeichnungen von Werkzeugen und Aufnahmen gilt es, einige Rahmenbedingungen einzuhalten. Standardmäßig wird ein Bild im 'Graphics Interchange Format' (GIF) mit 342x226 Bildpunkten erwartet. Die Werkzeugverwaltung unterstützt des weiteren die Formate JPEG (Join Expert Picture Group), BMP (Bitmap) und WMF (Windows Meta File). Unterscheidet sich die Bildgröße von den Standardabmessungen, so wird ein zu kleines Bild in der linken oberen Ecke der Anzeigefläche dargestellt, während bei einem zu großen Bild der überschüssige rechte und untere Rand abgeschnitten wird.

Für die Benennung der Bilder gilt es ebenfalls einige Konventionen einzuhalten, da die Werkzeugverwaltung für neu angewählte Werkzeugklassen automatisch nach Bildern mit entsprechenden Namen sucht. Wird zum Beispiel im Hauptfenster die Klasse 'TNutenfraeser' angewählt, so wird versucht, ein Bild namens „TNutenfraeser.gif“ im Standardverzeichnis zu finden. Kann keine entsprechende Datei gefunden werden, wird ein weißes Rechteck angezeigt und das Bild muss manuell ausgewählt werden.

Für die Ablage der Bilddateien existiert relativ zur Hauptanwendung ein Verzeichnis 'Pic'.

```
\ToolAdmin\Pic\..
```

Die Hauptaufgabe der Bilder ist die Erklärung der geometrischen Parameter, die für eine Aufnahme oder ein Werkzeug benutzt werden. Zur der Werkzeugklasse 'Schaffraeser' gehören zum Beispiel die Parameter 'InsertDiam', 'InsertLength', 'HolderDiam', 'TotalLength' und 'CornerRadius'. Das zugehörige Standardbild „Schaffraeser.gif“ wird in Abbildung 22 gezeigt:

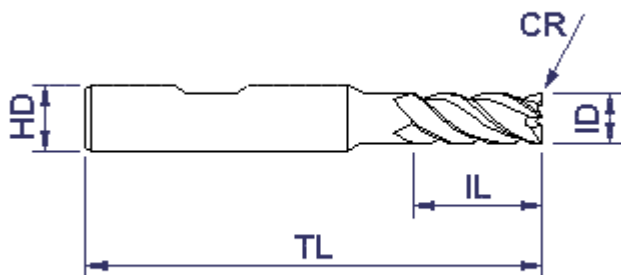


ABBILDUNG 22. Zeichnung zur Werkzeugklasse 'Schaffraeser'

5 Zusammenfassung und Ausblick

Ziel war es, eine Werkzeugverwaltung zu entwickeln, die den Bedürfnissen der Fräsimulation angepasst ist und neben der Verwaltung von Werkzeugen auch die Erzeugung von GD-Dateien erlaubt.

- Die entstandene Werkzeugverwaltung ermöglicht das Verwalten (also das Anlegen, Löschen und Ändern) von Werkzeugen, Werkzeugaufnahmen und zusammengesetzten Komplettwerkzeugen.
- Es ist möglich, sich in der Werkzeugverwaltung eine Vorschau der 3D-Werkzeuge anzeigen zu lassen, ohne die Fräsimulation zu starten.
- Die von der Simulation benötigten GD-Dateien können automatisch aus den gespeicherten Komplettwerkzeugen erstellt werden.
- Zu jeder Werkzeugklasse gibt es einige Beispieldatensätze, in der Datenbank, die als Muster für weitere Einträge dienen.

Um die Funktionsfähigkeit der Werkzeugverwaltung sicher zu stellen, wurde das System in der Firma an verschiedenen Arbeitsplätzen mit unterschiedlichen Fräsprojekten und von mehreren Mitarbeitern getestet.

Die Fortschritte des Projekts wurden über zwei Zwischenpräsentationen und eine Abschlusspräsentation den anderen Mitarbeitern bekanntgemacht.

Da alle Anforderungen an die Software erfüllt wurden, die Probeläufe erfolgreich verlaufen sind und der Auftraggeber mit den Ergebnissen zufrieden ist, ist dieses Projekt erfolgreich abgeschlossen.

Ausblick

Sobald die Simulation 'CNC Milling' vermarktet wird, soll die Werkzeugverwaltung als Zusatzmodul mit angeboten werden. Durch die aktuelle Entwicklung der Simulation ergeben sich einige Erweiterungsmöglichkeiten:

- Zur Zeit wird ein 'Setup Wizard' entwickelt, der die manuelle Konfiguration eines Fräsprojekts ersetzen soll. Da auch das Bereitstellen von Werkzeugen ein Teil der Konfiguration ist, soll die Werkzeugverwaltung in den Wizard integriert werden.
- Falls ein Kunde bereits eine Werkzeugverwaltung (z.B. Walter TDM) besitzt und zusätzlich die STA-Simulation erwirbt, soll eine Schnittstelle geschaffen werden, die einen Werkzeugtransfer von einem System zum Anderen ermöglicht.

Anhang

A Glossar

Bearbeitungszentrum	CNC-Werkzeugmaschine, die mehrere verschiedene Fertigungsarten beherrscht (z.B.: Fräs-Bearbeitungszentrum: Fräsen und Bohren)
Fremdschlüssel	Spalte in einer Tabelle, in der ein Primärschlüssel einer anderen Tabelle referenziert wird.
Halbzeug	Ein Rohteil (z.B. ein Stück abgesägtes Stangenmaterial), aus dem ein Fertigteil erzeugt werden soll.
Member	Diese Bezeichnung gibt an, dass eine Variable oder eine Funktion zu einer Klasse gehört.
Methode	Eine Funktion, die zu einer Klasse gehört.
NC-Steuerung	Numerical Control-Steuerung, für die Abarbeitung von Anweisungen an einer NC-Maschine.
Primärschlüssel	Dient zur eindeutigen Identifikation eines Tabelleneintrags.
Rotationskörper	Ein dreidimensionales Objekt, das durch die Rotation einer 2D-Kontur um eine Achse entsteht.
Rüstzeit	Zeit, die für das Umrüsten einer Maschine benötigt wird, z.B. für Werkzeugwechsel, Umspannungen des Werkstücks, u.ä..
Schneidplatten	Auswechselbare Schneiden, an Werkzeugen.
Spannmittel	Befestigungsmittel um Werkstücke in einer Werkzeugmaschine zu fixieren, z.B. Schraubstöcke oder Spannpratzen.
Spindeldrehzahl	Umdrehungen pro Minute einer Werkzeugspindel.
String	Ein Datentyp, der Zeichenketten beliebiger Länge aufnehmen kann.
Vorschubgeschwindigkeit	Geschwindigkeit in [m/min], mit der das Werkzeug verfahren wird.

Werkzeugmagazin Aufnahmevorrichtung für Werkzeuge innerhalb einer Werkzeugmaschine.

B Abkürzungsverzeichnis

API	Application Programmers Interface
CAD	Computer Aided Design
CAM	Computer Aided Manufacturing
CNC	Computerized Numerical Control
DXF	Data Exchange Format
IGES	Initial Graphics Exchange Specification
PPS-System	Produktions Planungs- und Steuerungssystem
VDAPS	Verband Deutscher Automobilhersteller Programm-Schnittstelle
VRML	Virutal Reality Modeling Language
XML	Extensible Markup Language

C Abbildungsverzeichnis

ABB. 1.	Bearbeitungszentrum DMU 80P der Firma Deckel Maho	5
ABB. 2.	Komponenten von 'CNC Milling'	7
ABB. 3.	konfigurierte Maschine.....	8
ABB. 4.	Vergleich verschiedener Werkzeugverwaltungen.....	14
ABB. 5.	Verschiedene Fräswerkzeuge	16
ABB. 6.	Kugelkopffräser mit Aufnahme aus der Simulation 'CNC Milling'	17
ABB. 7.	Werkzeuge	19
ABB. 8.	Werkzeugaufnahmen	19
ABB. 9.	Komplettwerkzeug	19
ABB. 10.	Radiuskorrektur	23
ABB. 11.	Anwendungsfalldiagramm	25
ABB. 12.	Datenbankkommunikation über ODBC	34
ABB. 13.	Datenmodell	36
ABB. 14.	Klassendiagramm der Hauptanwendung.....	38
ABB. 15.	Hauptdialog der Werkzeugverwaltung.....	39
ABB. 16.	Werkzeugauswahldialog.....	41
ABB. 17.	Dialog zur Auswahl von Komplettwerkzeugen	42
ABB. 18.	Dialog zur Verwaltung von Aufnahmen.....	43
ABB. 19.	Editierdialog (für Werkzeugmaterialien).....	44
ABB. 20.	Eingabemaske für Werkzeugmaterialien.....	44
ABB. 21.	2D-Kontur eines Komplettwerkzeugs	50
ABB. 22.	Zeichnung zur Werkzeugklasse 'Schaftfraeser'	53

D Quellen

Literatur

- [Balzert 99] Balzert, H. *Lehrbuch der Objektmodellierung*, Spektrum Akademischer Verlag, Heidelberg, 1999
- [Booch et al. 99] Booch, G., Rumbaugh J., Jacobson I. *Das UML-Benutzerhandbuch*, Addison Wesley Verlag, München, 1999
- [DIN 92] *DIN-Taschenbuch 167 - Fräswerkzeuge*, Beuth Verlag GmbH, Berlin, 1992
- [Fowler & Scott 2000] Fowler, M., Scott, K. *UML konzentriert*, Addison-Wesley Verlag, München, 2000
- [Oestereich 98] Oestereich, B. *Objektorientierte Softwareentwicklung*, R. Oldenburg Verlag München, 1998
- [Ousterhout 96] Ousterhout, John K., *Tcl and the Tk Toolkit*, Addison-Wesley Publishing Company, Massachusetts, 1996
- [Prosise 99] Prosise, J. *Windows-Programmierung mit MFC*, Microsoft Press Deutschland, Unterschleißheim, 1999
- [STA 99] *STA Fileformat Specification - Geometry Data - Turning*, STA GmbH, Berlin, 1999
- [STA 03] *Specification of the STA GD Fileformat*, STA GmbH, Berlin, 2003
- [Tschätsch 02] Tschätsch, H. *Praxis der Zerspantechnik*, Vieweg Verlag GmbH, Braunschweig, 2002

Internet

- [BAMiTec 03] BAMiTec AG, Grenchen, www.bamitec.ch, 2003
- [Bykov 03] Oleg Bykov, *Add GIF-animation to your MFC and ATL projects with the help of CPictureEx and CPictureExWnd*, www.codeproject.com/bitmap, 2003
- [CIM 03] CIM GmbH, Aachen, www.cimsource.com, 2003
- [Datos 03] DATOS Computer AG, Zürich, www.datos.ch, 2003
- [MySQL 03] MySQL GmbH, Nürtingen, www.mysql.de, 2003
- [Walter 03] Walter AG, Tübingen, www.walter-ag.de, 2003

E Beispiel zur Verwendung der Werkzeugverwaltung

Dieses Beispiel soll einen Überblick über die Funktionsweise der Werkzeugverwaltung geben. Es wird ein Komplettwerkzeug aus einem T-Nutenfräser und einer SK30 Aufnahme zusammengestellt und anschließend eine GD-Datei generiert.

1. Dies ist der Startbildschirm der Werkzeugverwaltung. Um ein Werkzeug auszuwählen, muss die Schaltfläche 'choose' im Bereich 'Tool' angeklickt werden.

The screenshot shows the 'ToolAdmin' software interface. The left sidebar contains the following fields:

- Toolclass: [edit]
- Description:
- Comment:
- Type of holder:
- Type of insert:
- Manufacturer: [edit]
- Insert material: [edit]
- Material comment:

The central area contains a grid of checkboxes, all labeled 'not used':

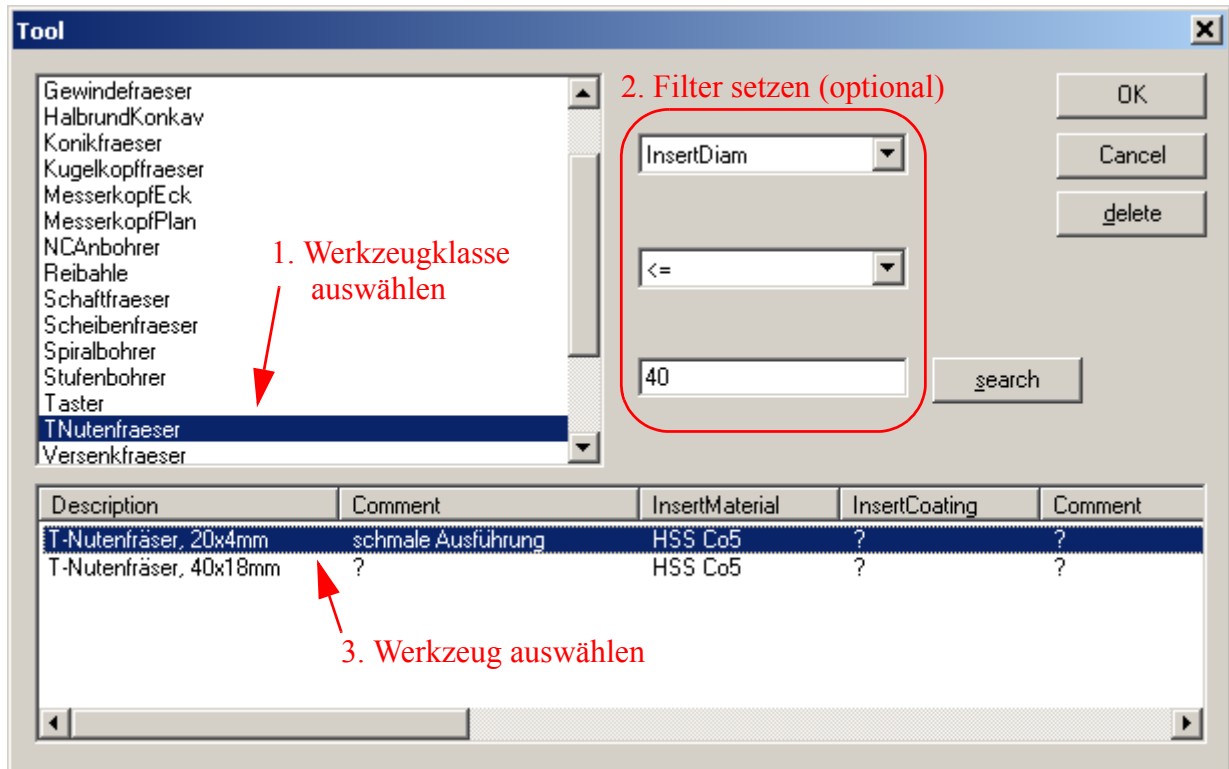
not used	<input type="checkbox"/>	not used	<input type="checkbox"/>
not used	<input type="checkbox"/>	not used	<input type="checkbox"/>
not used	<input type="checkbox"/>	not used	<input type="checkbox"/>
not used	<input type="checkbox"/>	not used	<input type="checkbox"/>
not used	<input type="checkbox"/>	not used	<input type="checkbox"/>

The right sidebar contains the following sections:

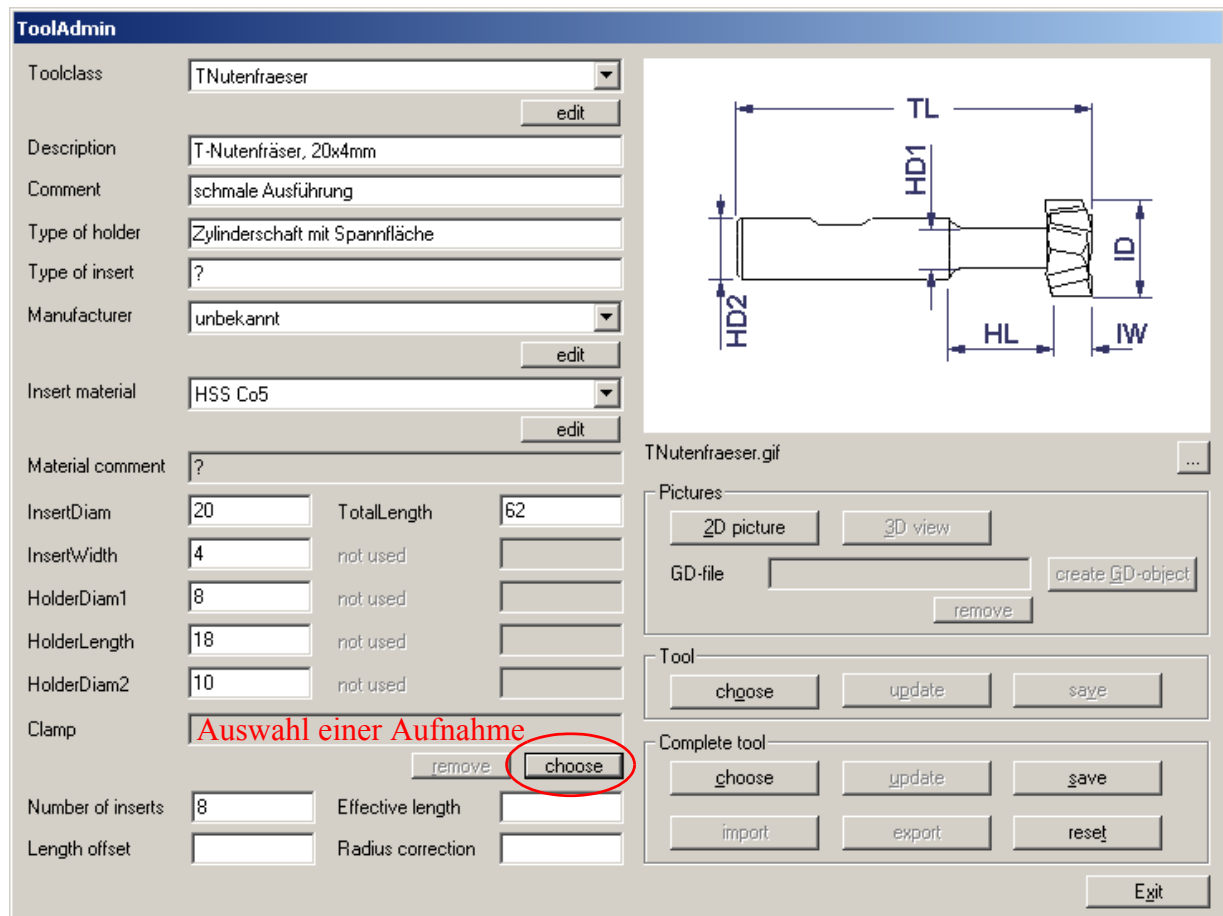
- Pictures:
- GD-file:
- Tool:
- Complete tool:

An 'Exit' button is located at the bottom right of the interface.

2. Auswahl eines T-Nutenfräasers



3. Die Werkzeugdaten wurden in den Hauptdialog übernommen



4. Auswahl einer Aufnahme

Description	ConeTy...	ConeNorm	ToolClampType	Comment	Dim1	Dim2
Bohrfutter 3-16mm, SK40 DIN69871	SK 40	DIN 69871	Bohrfutter	Zum Spannen v...	47	100
Bohrfutter klein, SK30 DIN2080	SK 30	DIN 2080	Bohrfutter	Zum Spannen v...	30	40
Bohrfutter spezial, SK30 DIN69871	SK 30	DIN 69871	Bohrfutter	Zum Spannen v...	50	55
Fräsespannfutter 06mm, SK30 DIN2080	SK 30	DIN 2080	Fräsespannfutter	Für Werkzeuge ...	25	50
Fräsespannfutter 06mm, SK30 DIN69871	SK 30	DIN 69871	Fräsespannfutter	Für Werkzeuge ...	25	40
Fräsespannfutter 08mm, SK30 DIN69871	SK 30	DIN 69871	Fräsespannfutter	Für Werkzeuge ...	30	40
Fräsespannfutter 10mm, SK30 DIN2080	SK 30	DIN 2080	Fräsespannfutter	Für Werkzeuge ...	35	50
Fräsespannfutter 10mm, SK30 DIN69871	SK 30	DIN 69871	Fräsespannfutter	Für Werkzeuge ...	35	40

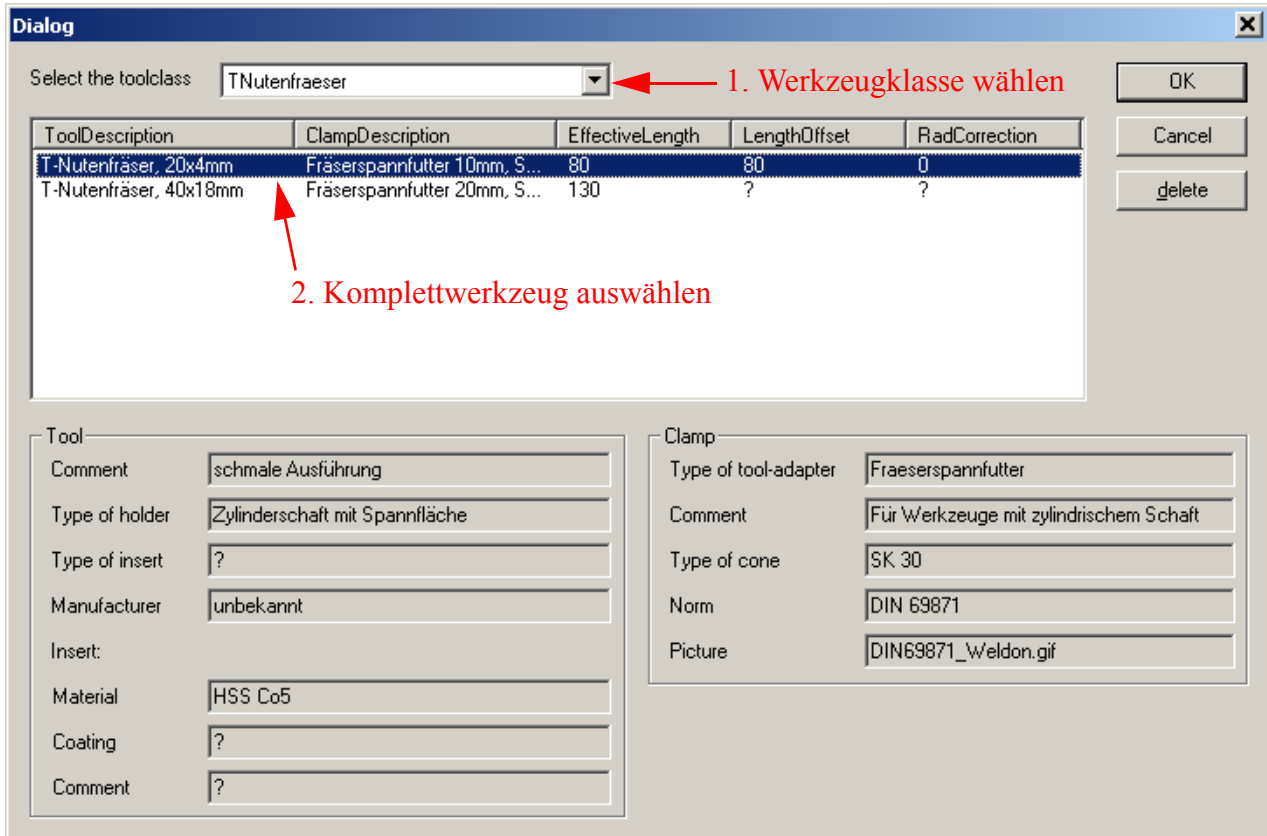
5. Restliche Parameter eintragen und GD-Datei erzeugen

3. Vorschau wird angezeigt

2. GD-Datei erzeugen

1. Parameter eingeben

Alternativ zur Zusammenstellung einzelner Werkzeuge lässt sich auch gleich nach Programmstart ein Komplettwerkzeug auswählen. Dazu muss im Hauptbildschirm die Schaltfläche 'choose' im Bereich 'Complete tool' angeklickt werden, wonach der folgende Auswahldialog erscheint:



Nach dem Beenden des Dialogs mit 'OK' (oder einem Doppelklick auf ein Werkzeug) erscheint wieder der Hauptdialog mit allen Daten zum ausgewählten Werkzeug. Die GD-Datei kann jetzt direkt erstellt werden.

Diese Variante entspricht genau dem in Kapitel 3.2.1 beschriebenen Anwendungsfall 5, während die erste Variante der Anwendungsfall 4.1 ist.