



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bachelorarbeit

Andrej Markus

Design und Entwicklung eines Wellen-Konfigurators in PTC Creo

*Fakultät Technik und Informatik
Department Fahrzeugtechnik und Flugzeugbau*

*Faculty of Engineering and Computer Science
Department of Automotive and
Aeronautical Engineering*

Andrej Markus

**Design und Entwicklung eines Wellen-
Konfigurators in PTC Creo**

Bachelor-/Masterarbeit eingereicht im Rahmen der Bachelorprüfung

im Studiengang Mechatronik
am Department Maschinenbau und Produktion
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Erstprüfer: Prof. Dr.-Ing. Hans-Joachim Schelberg
Zweitprüfer: Prof. Dr.-Ing. Hartmut Noack

Abgabedatum: 19.08.2016

Zusammenfassung

Andrej Markus

Thema der Bachelorthesis

Design und Entwicklung eines Wellen-Konfigurators in PTC Creo

Stichworte

Wellen-Konfigurator, PTC Creo, PTC Creo VB API, Software, Implementierung, Konstruktion

Kurzzusammenfassung

Diese Arbeit umfasst die Entwicklung und Implementierung eines Wellen-Konfigurators für die schnelle Erzeugung der Geometrie einer mechanischen Welle im CAD-System PTC Creo.

Andrej Markus

Title of the paper

Design and developing of wave configuration in PTC Creo

Keywords

Wave configuration, PTC Creo, PTC Creo VB API, software, implementation, construction

Abstract

Inside this report is the development and implementation of wave configuration for fast production of geometry from a mechanical wave in the CAD system PTC Creo.

Inhaltverzeichnis

1	Einleitung.....	7
1.1	Ziel der Arbeit.....	7
1.2	Gliederung der Arbeit.....	7
2	Anforderungen und Stand der Technik.....	8
2.1	Welle: Definition und Funktion	8
2.2	Anforderungen an die Wellenkonstruktion	8
2.3	Verfügbare Lösungen	9
2.3.1	Wellen-Generator in Autodesk Inventor	9
2.3.2	eAssistant von GWJ	11
2.3.3	SolidGenius von MIP Software	12
2.3.4	Fazit	13
3	Generische Produktarchitektur.....	13
3.1	Arten von Absätzen	14
3.1.1	Hohlwelle.....	14
3.1.2	Zylinder.....	15
3.1.3	Kegel	16
3.1.4	Keilwelle	17
3.1.5	Passverzahnung mit Evolventenflanken.....	18
3.1.6	Polygonprofil P3G.....	19
3.1.7	Polygonprofil P4C	20
3.2	Weitere Elemente an Wellen	21
3.2.1	Zentrierbohrung	21
3.2.2	Radialbohrung	22
3.2.3	Fase.....	23
3.2.4	Rundung	24
3.2.5	Gewinde.....	25
3.2.6	Gewindefreistich	26
3.2.7	Freistich	27
3.2.8	Passfedernut Form A	28
3.2.9	Nut für Sicherungsringe.....	29
3.2.10	Nut für Sicherungsblech	30
3.3	Schwierigkeiten	31
4	Implementierung der Funktionen des Wellen-Konfigurators in VB API.....	32
4.1	Allgemeine Informationen zu VB API	32

4.2	Programmierungsumgebung	32
4.3	Applikationshierarchie.....	33
4.4	Funktionen des Wellen-Konfigurators	34
4.4.1	Verbindung herstellen	35
4.4.2	Verbindung trennen	35
4.4.3	Wellenabsatz löschen.....	35
4.4.4	Parameter ändern	38
4.4.5	Aufruf eines Modells	39
4.4.6	Aufruf einer Schablone.....	39
4.4.7	Eine Komponente in Baugruppe einfügen	39
4.4.8	Modelle in einer Baugruppe umordnen.....	42
4.4.9	Alle Modelle in der Baugruppe auflisten.....	42
4.4.10	Alle Modelle in einer Liste darstellen	43
4.4.11	Ein Modell aktualisieren	43
4.4.12	Ein Feature eines Modells unterdrücken	43
4.4.13	Ein Feature löschen	43
4.4.14	Ein Feature wieder sichtbar machen.....	44
4.4.15	UDF in ein Modell einfügen.....	44
4.4.16	Zeichnung	45
4.4.17	Gewindedarstellungen wechseln	46
5	Dialogorientierte Benutzeroberfläche	48
6	Fazit und Ausblick.....	49



19.08.2016

AUFGABENSTELLUNG DER BACHELORARBEIT

von Andrej Markus

Thema: Design und Entwicklung eines Wellen-Konfigurators in PTC Creo

Hintergrund:

Die Konstruktion mechanischer Wellen ist ein wichtiger Bestandteil jedes Studienganges des Maschinenbaus. Hierbei sollte jedoch die konstruktive Auslegung und Gestaltung der Welle im Vordergrund stehen und nicht die Erzeugung der zugehörigen Geometrie im CAD-System. In dieser Arbeit soll daher in PTC Creo ein sog. Wellen-Konfigurator entwickelt werden, der auf Basis einiger weniger Anforderungs- und Auslegungskriterien teil- oder auch vollautomatisch eine gewünschte mechanische Welle im CAD-System generiert. Damit soll zugleich die Grundlage für spätere Konfiguratoren, bspw. Getriebe oder Kupplungen, geschaffen werden.

Aufgabenstellung:

Aufbauend auf einer intensiven Marktrecherche sollen zunächst die methodischen und programmiertechnischen Grundlagen erarbeitet werden. Im zweiten Schritt sollen die Grundfunktionalitäten des Wellen-Konfigurators inkl. der erforderlichen Bedienfunktionalitäten unter PTC Creo implementiert werden.

Im Einzelnen ergeben sich daraus folgende Arbeitspakete:

- Recherche über
 - Anforderungen an die Wellenkonstruktion im CAD-System
 - Marktverfügbare Lösungen
- Definition der generischen Produktarchitektur des Wellen-Konfigurators mit den benötigten Modulen (Bauteilen, Baugruppen) und deren Schnittstellen (Geometrie, Beziehungen)
- Ableitung und Implementierung eines entsprechend parametrisierten CAD-Modells
- Entwurf und Implementierung der für die Bedienung des Konfigurators erforderlichen Funktionen in Form einer dialogorientierten Benutzeroberfläche
- Dokumentation der implementierten Lösung
- Empfehlungen für die Weiterentwicklung

1. Betreuer: Prof. Dr.-Ing. Hans-Joachim Schelberg
Institut für Konstruktion und Produktentwicklung
Berliner Tor 21, 20099 Hamburg

2. Betreuer: Prof. Dr.-Ing. Hartmut Noack
Institut für Konstruktion und Produktentwicklung
Berliner Tor 21, 20099 Hamburg

1 Einleitung

Heutzutage sind die Konstruktion und die Fertigung im Maschinenbau ohne die Unterstützung von 3D-CAD-Systemen unvorstellbar geworden. Diese Systeme werden immer umfangreicher und bewältigen immer größere Aufgaben. Mit ihrer Hilfe erstellt man komplexe 3D-Modelle, technische Zeichnungen, stellt sämtliche Berechnungen an, animiert und/oder simuliert seine Produkte, führt die Dokumentation uvm. Einige Systeme bieten sogar spezielle Tools an, die vollständige 3D-Modelle von normgerechten Maschinenteilen (wie z.B. Zahnräder, Lager, Federn, mechanische Wellen etc.) vollautomatisch generieren und somit den Nutzern helfen, deren Produkte schneller und effizienter zu entwickeln bzw. zu produzieren.

1.1 Ziel der Arbeit

Die Konstruktion mechanischer Wellen ist ein wichtiger Bestandteil jedes Studienganges des Maschinenbaus. Hierbei sollte jedoch die konstruktive Auslegung und Gestaltung der Welle im Vordergrund stehen und nicht die Erzeugung der zugehörigen Geometrie im CAD-System. In dieser Arbeit soll daher in PTC Creo, das aktuell an der Hochschule für Angewandte Wissenschaften Hamburg für die Lehr- bzw. Lernzwecke eingesetzt wird, ein sog. Wellen-Konfigurator entwickelt werden. Er soll auf Basis einiger weniger Anforderungs- und Auslegungskriterien teil- oder auch vollautomatisch eine gewünschte mechanische Welle im CAD-System generieren. Den Studierenden der HAW wird damit ein Werkzeug geboten, das sie bei der Erzeugung der Geometrie einer Welle entlastet und auf keinen Fall ihre Kreativität bei der Gestaltung ersetzt.

Diese Arbeit kann zugleich als die Grundlage für spätere Konfiguratoren, bspw. Getriebe oder Kupplungen, dienen. An diesem Beispiel können alle Studierenden sich die Fähigkeiten aneignen, die ihnen ermöglichen werden, selbst solche Modellierfunktionen zu entwerfen. Dafür werden natürlich die Vorkenntnisse in der Konstruktion mit PTC Creo und die Programmierkenntnisse in VBA vorausgesetzt. Mit diesen Fähigkeiten kann man dann alle wiederkehrende Aufgaben in der Konstruktion automatisieren, was sicherlich gewisse Vorteile wie im Studium so auch im späteren Berufsleben mit sich bringen wird.

1.2 Gliederung der Arbeit

Die Arbeit ist in sechs Kapitel unterteilt. Die Themen und ihre kurzen Zusammenfassungen:

1. Einleitung – Führt in das Thema und das Ziel der Arbeit ein.
2. Anforderungen und Stand der Technik – Definiert die Anforderungen, die an den Wellen-Konfigurator gestellt werden, und gibt einen Überblick über verfügbare Produkte auf dem Markt.
3. Generische Produktarchitektur – Erläutert Geometrien, aus denen eine mechanische Welle bestehen kann.

4. Implementierung der Funktionen des Wellen-Konfigurators in VB API – Gibt allgemeine Informationen über die Programmierumgebung und beschreibt alle Modellierungsfunktionen.
5. Dialogorientierte Benutzeroberfläche – Gibt einen Überblick über die Bedienungselemente des Wellen-Konfigurators
6. Fazit und Ausblick - Fasst die innerhalb der Arbeit erhaltenen Erkenntnisse zusammen, bewertet diese Anhand der Zielsetzung und gibt einen kurzen Ausblick über eine mögliche Weiterentwicklung.

2 Anforderungen und Stand der Technik

2.1 Welle: Definition und Funktion

Bei einer mechanischen Welle handelt es sich um ein Maschinenelement in Form einer zylindrischen Stange. Ihre Hauptfunktion besteht darin, die Drehbewegungen und die Drehmomente zu übertragen. Dabei werden sie meistens auf Torsion und zusätzlich durch die Querkräfte auf Biegung beansprucht. Diese Belastungen werden von Konstrukteuren ermittelt und bei der Auslegung berücksichtigt. Demnach wird die Welle entsprechend gestaltet.

Auf die maßgeblichen Wellenelemente, die bei der Formgestaltung einer Welle die Rolle spielen und die jeweilige Funktion übernehmen, wird es genauer in folgenden Unterkapiteln eingegangen.

2.2 Anforderungen an die Wellenkonstruktion

Selbstverständlich erwartet man und strebt danach, dass sein konstruiertes Modell im CAD-System detailgetreu und übereinstimmend mit dem späteren Werkstück sein soll. Wie schon davor erwähnt wurde, ermöglichen die heutigen CAD-Systeme mit der aktuellen Hardware praktisch jede Formgestaltung. Es ist also möglich jede beliebige mechanische Welle im CAD, unabhängig davon ob man es manuell oder automatisch macht, umzusetzen. Da stellen sich die Fragen, welche Anforderungen man setzen möchte, wie weit man die gesetzten Ziele verfolgen kann und ob es überhaupt sinnvoll und notwendig ist, dass die Abbildungen hundertprozentig der Wirklichkeit entsprechen.

Die wichtigsten Funktionen, die der Wellen-Konfigurator besitzen soll:

- Generierung realitätsnaher Geometrie

- Das Modell der Welle soll veränderbar sein. D.h. wenn man die Welle konventionell von Grund auf neu modelliert, erstellt man zuerst die Skizze vom Umriss der ganzen Welle, die schließlich um eine Achse 360° „gedreht“ wird. Die Problematik bei dieser Vorgehensweise ist, dass die Form der Welle unveränderbar bleibt und falls man sie ändern möchte, muss die vorhandene Welle gelöscht und die neue wieder modelliert werden. Die Lösung für das Problem sieht wie folgt aus: man setzt eine Welle aus vielen verschiedenen vordefinierten Absätzen zusammen. Die einzelnen Absätze werden löschar oder ersetzbar sein, was die flexible Formgestaltung gewährleisten wird.
- Nachdem der Grundkörper fertig sein wird, kann man die weiteren dazugehörigen Elemente (wie Freistich, Fase, Gewinde etc.) einfügen.
- Idealerweise wäre es, wenn der Konfigurator auch eine Zeichnung ableiten würde.
- Anschließend wird eine ergonomische dialogorientierte Benutzeroberfläche für die Bedienung des Konfigurators entworfen. Die Anwender sollen interaktiv und ohne besondere Einarbeitung in der Lage sein den Konfigurator zu bedienen.

2.3 Verfügbare Lösungen

Die Recherche nach den verfügbaren Lösungen eines fertigen Wellen-Konfigurators hat ergeben, dass der Wunsch nach der automatisierten Konstruktion mechanischer Wellen schon längst dafür viele verschiedene Lösungen geliefert hat. Einige davon werden hier vorgestellt und kurz zusammengefasst:

2.3.1 Wellen-Generator in Autodesk Inventor

Einer der bekanntesten Wellen-Konfiguratoren ist der Wellen-Generator in Autodesk Inventor.

„Mit dem Wellen-Generator können Sie die Form einer Welle konstruieren, Belastungen und Auflager sowie andere Berechnungsparameter hinzufügen und berechnen. Sie können eine Festigkeitskontrolle durchführen und die Welle in Autodesk Inventor erstellen. Eine Welle besteht aus einzelnen Elementen (Zylinder, Kegel und Polygon) mit Objekten (Fasen, Rundungen, Einstichen usw.).

Mit dem Wellen-Generator können Sie folgende Aufgaben durchführen:

- *Konstruieren und Einfügen von Wellen mit einer unbegrenzten Anzahl an Elementen (Zylinder, Kegel, Polygone) und Objekten (Kehlnähte, Fasen, Gewinde usw.)*
- *Konstruieren einer Hohlwelle*
- *Einfügen von Objekten (Fasen, Kehlnähte, Gewinde) in Innenhohlräume*

- *Teilen des Wellenzylinders und Beibehalten der Länge des Wellenelements*
- *Speichern von Wellen in der Vorlagenbibliothek*
- *Hinzufügen einer unbegrenzten Anzahl von Belastungen und Auflagern zur Wellenkonstruktion „[01]*

Hier sind einige Funktionsbeispiele des Wellen-Generators von Autodesk Inventor aufgelistet:

- *„Kantenobjekte*

Objekte, die nur auf der Wellenkante platziert werden können, z. B. Fasen, Kehlnähte, Sicherungsmuttern, Gewinde. Sie können nur ein Element auf der Wellenkante platzieren.

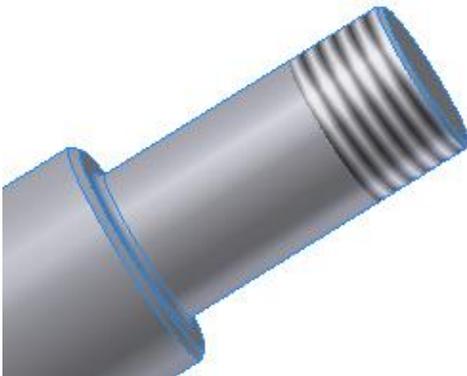


Abbildung 2.1: Wellen-Generator: Fase, Gewinde [01]

- *Mittlere Objekte*

Objekte, die auf dem Wellenelement platziert werden können, z. B. Keilnuten, Sicherungsringe, Kerben usw. Sie können eine unbegrenzte Anzahl dieser Elemente auf einem Wellenschnitt platzieren.



Abbildung 2.2: Wellen-Generator: Keilnut, Nut für Sicherungsring [01]

- *Positionsmarkierung*

Definiert Punkte auf einem Element, auf dem die Belastungen und Auflager positioniert werden. Sie können eine Kraft oder ein Auflager durch Drücken der ALT-Taste und Ziehen verschieben und einer anderen Positionsmarkierung zuweisen.

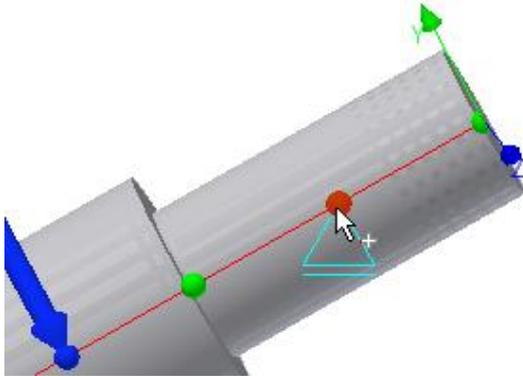


Abbildung 2.3: Wellen-Generator: Punkte für Auflager und Belastungen [01]

- *Grafisches Vorschauen*

Wenn eine Welle mit dem Wellengenerator konstruiert wird, ist die grafische 3D-Vorschau (mit allen Elementen, Objekten und 3D-Griffen) im Autodesk Inventor-Fenster verfügbar. Wenn Sie auf die Registerkarte Berechnung wechseln, wird die grafische Vorschau im Autodesk Inventor-Fenster und auf der Registerkarte Berechnung in die Vorschau der Belastungen und Auflager geändert.“ [01]

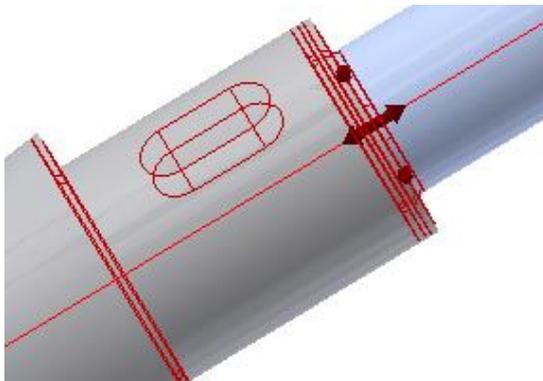


Abbildung 2.4: Wellen-Generator: 3D-Vorschau [01]

2.3.2 eAssistant von GWJ

Eine andere Lösung präsentiert das Unternehmen GWJ Technology GmbH mit seinem Plug-In Modul eAssistant vor, welches mit vielen CAD-Systemen kompatibel ist, darunter auch mit PTC Creo. Für Creo bietet es den folgenden Programmumfang an:

„Derzeit können folgende Teile auf Basis der eAssistant-Berechnungen in der 3D-Umgebung von Pro/ENGINEER erzeugt werden:

- geradzahnte Stirnräder (Außen- und Innenverzahnung)
- Kerbverzahnungen, Keilwellen (Welle/Nabe)
- Wellenkonturen (Außen-/Innenzylinder, Außen-/Innenkonus)

Außerdem:

- Speicherung der Berechnungsdaten im 3D-Modell
- Platzierung von Herstelldaten für Stirnräder in 2D-Zeichnungsumgebung
- Verzahnung kann auf einem bestehenden Teil erzeugt werden“ [02]

Das Tool unterstützt seine Anwender nicht nur bei der Erzeugung der Geometrie im jeweiligen CAD-System, sondern übernimmt sämtliche Berechnungen bei der Auslegung einer Welle. Auf Basis der ermittelten Daten werden dann entsprechende Modelle abgeleitet.

Von diesem Tool werden nur die Wellenabsätze und die Elemente einer Welle-Nabe Verbindung modelliert. Alle anderen Wellenelemente (Gewinde, Fase, Freistiche usw.) müssen konventionell eingefügt werden.

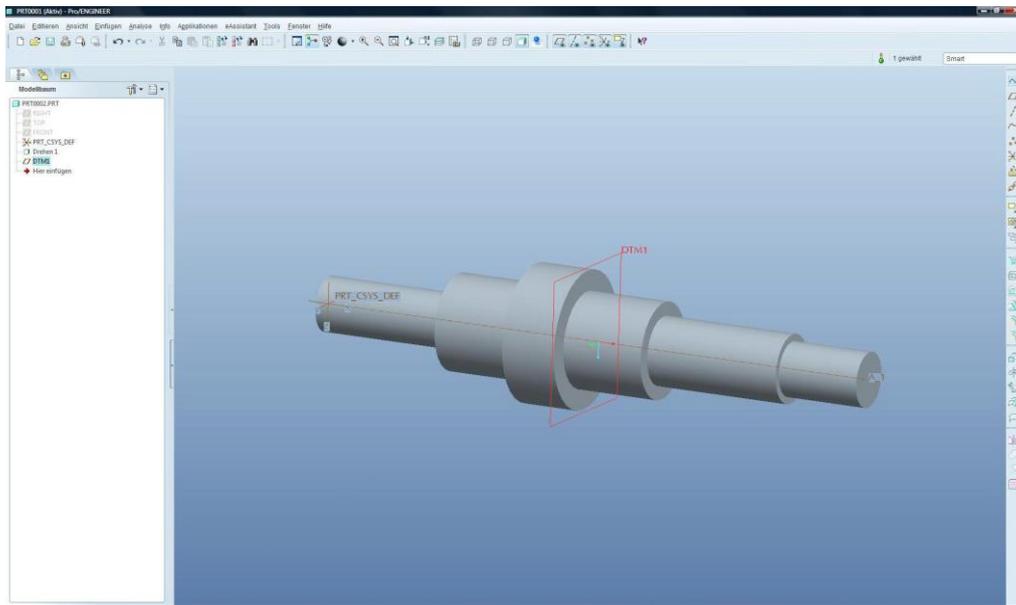


Abbildung 2.5: Erzeugung einer Welle von eAssistant in Creo [02]

2.3.3 SolidGenius von MIP Software

Ein etwas kleinerer Vertreter von Wellen-Konfiguratoren ist die Software SolidGenius von MIP. Dieses Tool wurde nur für PTC Creo Elements/Direct entwickelt. Es führt keine Berechnungen durch und modelliert keine Wellen. Mit diesem Werkzeug ist man aber in der Lage sehr schnell und unkompliziert an einer vorhandenen Welle viele Wellenelemente zu

erzeugen. So lassen sich z.B. Passfedern, Stifte, Schrauben, Sicherungsringe usw. einfügen. Zu diesen Normteilen werden auch automatisch dazugehörige Nuten bzw. Bohrungen erstellt.

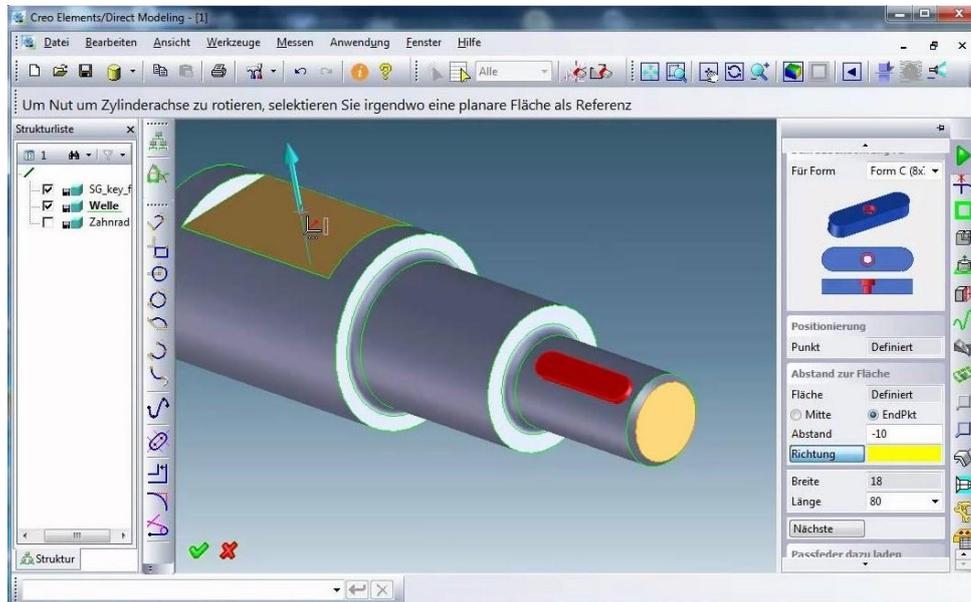


Abbildung 2.6: Erzeugung einer Passfeder in SolidGenius [03]

2.3.4 Fazit

Der Wellen-Generator von Autodesk Inventor entspricht mit seinem Funktionsumfang den gestellten Anforderungen am besten. Daher wird es im Verlauf der Entwicklung von dem Wellen-Konfigurator als eine ideale Vorlage genommen.

3 Generische Produktarchitektur

Die Gestaltung einer Welle ist vom Einsatz abhängig und sehr vielfältig. In diesem Kapitel wird nur auf die Fälle eingegangen, die in Lehrbüchern vorkommen und/oder im Studium behandelt werden. Dabei werden die maßgeblichen Komponenten und deren Funktion einer Welle betrachtet. Da es vorrangig um die Funktionalität des Wellen-Konfigurators gehen soll, sind die weitgehenden Betrachtungen (Auslegung/Berechnung) aus entsprechender Literatur zu entnehmen.

Wenn man unterschiedliche Wellen betrachtet, erkennt man, dass jede von ihnen eine Form und verschiedene Merkmale/Eigenschaften hat. Die Form wird durch unterschiedliche Absätze bestimmt. Jeder dieser Absätze übernimmt eine Funktion, die durch die Ergänzung von weiteren Elementen verändert werden kann.

Bei dieser Arbeit werden insgesamt sieben Arten von Absätzen und zehn weitere Elemente verwendet, die später im Einzelnen beschrieben und erklärt werden.

Um eine Geometrie in CAD-System zu erzeugen muss man großen Aufwand mit der Berechnung dieser Geometrie betreiben. Da es bei Absätzen und weiteren Elementen um die Normteile handelt, kann man für sie jeweils eine sog. Schablone (Template) konventionell vorfertigen. Die Absätze werden als Festkörper (Solid) modelliert und gespeichert. Für die weiteren Elemente, die nur als Ergänzung dienen sollen, gibt es in PTC Creo eine Möglichkeit sie als sog. UDFs (User Defined Features) zu speichern.

Damit man die Schablonen später nach Bedarf ändern kann, müssen sie parametrisch konstruiert werden. D. h. also, der Wellen-Konfigurator soll in der Lage sein, die Templates aufzurufen und ihnen neue Parameter zu übergeben. Die einzelnen Absätze mit oder ohne Ergänzungselemente werden in einer Baugruppe zu einer Welle zusammengebaut. Die größte Herausforderung dabei ist es, eine Baugruppe zu schaffen, welche bei Änderungen der Bauteile stabil bleibt. Wenn die Abmessungen eines Bauteils geändert werden oder ein Bauteil entfernt wird, kann es dazu führen, dass Creo die Körpergeometrien nicht mehr erkennt und die Abhängigkeiten zu dieser Geometrie verliert, was schließlich in einem Zusammenbruch der gesamten Konstruktion enden wird. Der Zusammenbau sollte also frei von den Referenzen auf die Körpergeometrien sein. Um Änderungen zuverlässig umzusetzen müssen separate Ebenen erstellt werden. Auf diese erstellten Ebenen, die als Grenzen oder Positionen eines Bauteils dienen, werden die Referenzen gesetzt.

Im Folgenden werden die verwendeten Arten von Absätzen und die weiteren Elemente präsentiert.

3.1 Arten von Absätzen

3.1.1 Hohlwelle

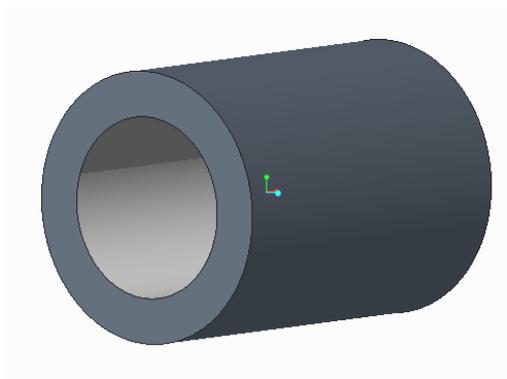


Abbildung 3.1: 3D-Modell: Hohlprofil

Der Hohlprofil kommt zum Einsatz für die Modellierung von Hohlwellen. Die wichtigste Funktion dafür ist die Materialeinsparung und damit die Gewichtsreduzierung einer Welle.

Ein großer Vertreter dieser Wellenart ist die Kardanwelle, die auch zum Teil zur Gelenkwellen gehört.

Parameter für den Wellen-Konfigurator:

- D_i : Innendurchmesser
- D_a : Außendurchmesser
- L : Länge des Absatzes

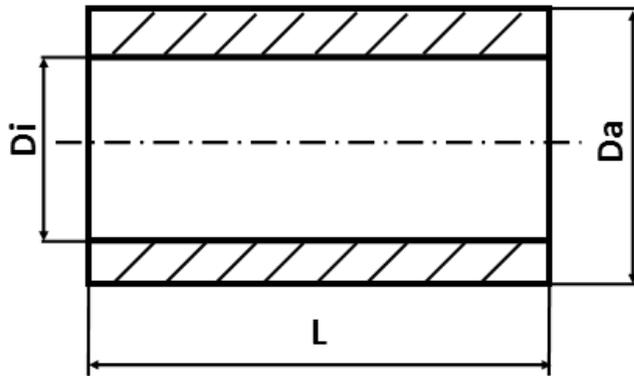


Abbildung 3.2: Skizze: Hohlprofil

3.1.2 Zylinder

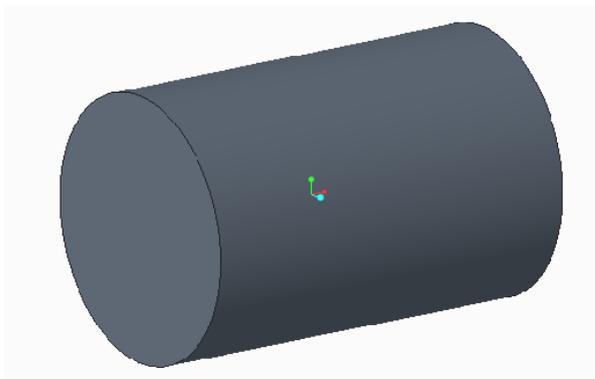


Abbildung 3.3: 3D Modell: Zylinder

Der Zylinder gehört zu den Rundprofilen und ist die am häufigsten verwendete Form für einen Absatz. Es liegt daran, dass man ihn sehr vielseitig einsetzen kann:

1. zur Aufnahme von Lagern
2. zur Drehmoment-Übertragung in Verbindung mit z.B. einer Passfeder bei einer formschlüssigen Welle-Nabe Verbindung
3. zur Drehmoment-Übertragung bei einer kraftschlüssigen Welle-Nabe Verbindung
4. zur Drehmoment-Übertragung bei einer stoffschlüssigen Verbindung
5. zur Kraftflussleitung usw.

Parameter für den Wellen-Konfigurator:

- L: Länge
- D: Durchmesser



Abbildung 3.4: Skizze: Zylinder

3.1.3 Kegel



Abbildung 3.5: 3D Modell: Kegel

Das Kegelfprofil gehört auch wie der Zylinder zu den Rundprofilen. Seine Funktion ist die Drehmoment-Übertragung. In Verbindung mit einer Passfeder kann der Kegel bei einer formschlüssigen Welle-Nabe Verbindung oder alleine bei einer kraftschlüssigen Welle-Nabe Verbindung eingesetzt werden. Sein Vorteil gegenüber dem Zylinder ist es, dass durch seine Form die Nabe nur von einer Seite gesichert werden muss.

Parameter für den Wellen-Konfigurator:

- L: Länge
- D1: Durchmesser auf einer Seite
- D2: Durchmesser auf der anderen Seite

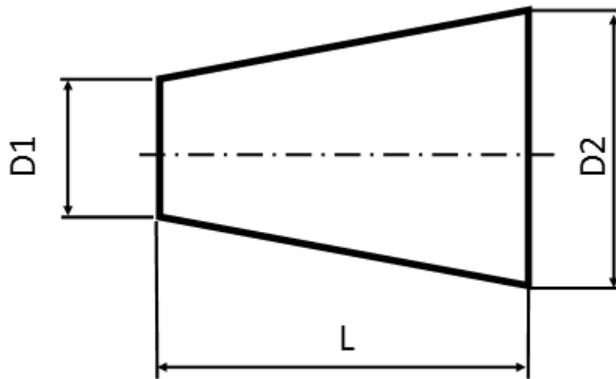


Abbildung 3.6: Skizze: Kegel

3.1.4 Keilwelle

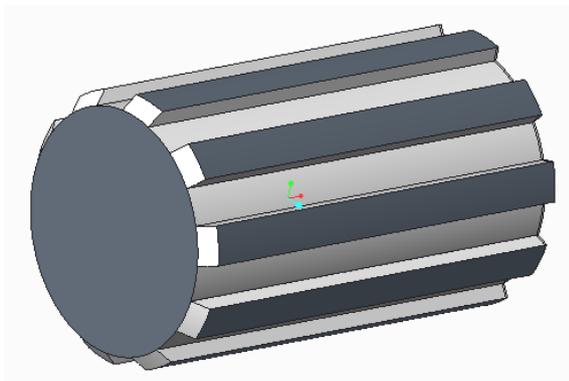


Abbildung 3.7: 3D Modell: Keilwelle

Die Keilwelle gehört zu den sog. Vielnutprofilen und zur formschlüssigen Welle-Nabe Verbindung. Durch die hohe Anzahl der Keile werden damit sehr große Drehmomente übertragen. Die Anzahl der Keile und alle Bemaßungen sind nach DIN ISO 14 genormt. Die Flanken sind radial und parallel zu einander zu konstruieren. Bei der Zeichnung hat dieses Profil auch die genormte Bezeichnung, die sich aus den folgenden Teilen zusammensetzt:

1. der Norm
2. einem Kennbuchstaben für Welle (W)
3. der Anzahl der Keile
4. einem Innendurchmesser
5. einem Außendurchmesser

So kann eine Bezeichnung beispielsweise aussehen: „DIN ISO 14 – W 10x82x88“. Die Maße werden der Tabelle aus dem Tabellenbuch Metall entnommen und im Wellen-Konfigurator gespeichert. So müssen die Anwender sie nicht mehr eintragen, sondern lediglich nur die Bezeichnung auswählen. Nur die Länge des Absatzes und die Länge der gefrästen Nut, falls sie nicht durchgehend ist, müssen immer separat eingegeben werden.

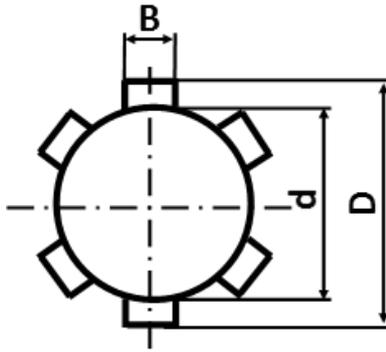


Abbildung 3.8: Skizze: Keilwellenprofil

3.1.5 Passverzahnung mit Evolventenflanken

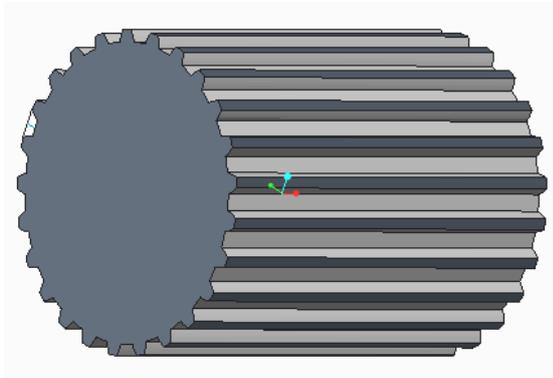


Abbildung 3.9: 3D Model: Zahnwelle mit Evolventenflanken

Die Zahnwelle mit Evolventenflanken gehört zu den formschlüssigen Welle-Nabe Verbindungen. Im Vergleich zu Keilwellen besitzt sie mehr Zähne. Dadurch sinkt die Kerbwirkung der Welle, was noch größere Drehmoment-Übertragung beim gleichen Durchmesser erlaubt. Die Zahnwelle ist nach DIN 5480 genormt. Ihre Abmessungen findet man in der Tabelle 12-4 b) in RoMa15.

Bei der Zeichnung hat auch dieses Profil die genormte Bezeichnung. Sie besteht aus:

1. der Normnummer
2. einem Kennbuchstaben für Welle (W)
3. dem Bezugsdurchmesser
4. dem Modul
5. der Zähnezahl
6. der Toleranzklasse

So kann eine Bezeichnung beispielsweise aussehen: „DIN 5480 – W 50x2x24x8f“.

Auch hier müssen die Anwender nur die Bezeichnung auswählen und die Länge des Absatzes eingeben.

3.1.6 Polygonprofil P3G

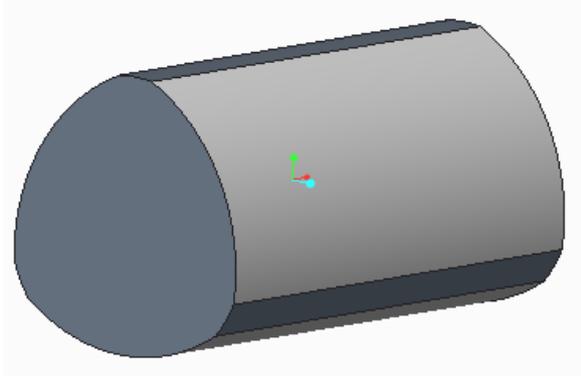


Abbildung 3.10: 3D Model: Polygonprofil P3G

Das Polygonprofil P3G ist ein Unrundprofil. Die Wellen mit diesem Profil finden in vielen Bereichen ihren Einsatz, z.B. in Werkzeugmaschinen, Kraftfahrzeugen, Elektroindustrie. Sie gehören zur formschlüssigen Welle-Nabe Verbindung und bieten die höchste Drehmoment-Übertragung von allen. Durch die sanften Übergänge der Flächen ist die Kerbwirkung praktisch zu vernachlässigen. Der Vorteil dieses Profils ist es, dass ihm stoßartige Drehmomente keine Probleme darstellen. Das Polygonprofil P3G ist nach DIN 32711 genormt. Die Bezeichnung setzt sich zusammen aus:

1. der Normnummer
2. einem Kennbuchstaben (hier: A für Welle)
3. der Profilbezeichnung
4. dem Durchmesser d_1

Ein Beispiel dafür: „DIN 32711 – A P3G 35“.

Parameter für den Wellen-Konfigurator:

- d_2 : Durchmesser 2
- d_3 : Durchmesser 3
- L: Länge des Absatzes

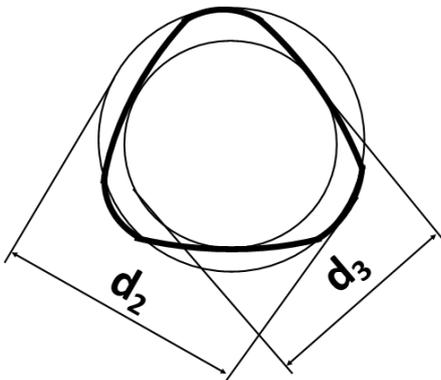


Abbildung 3.11: Skizze: Polygonprofil P3G

3.1.7 Polygonprofil P4C

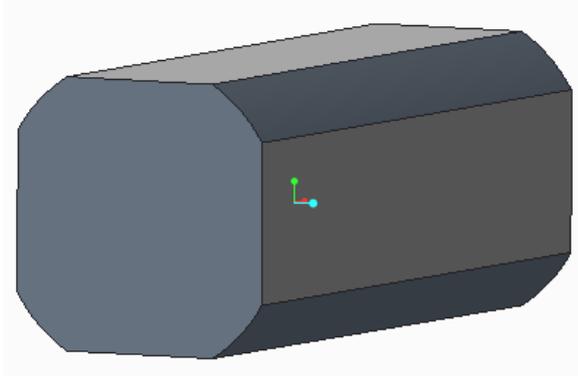


Abbildung 3.12: 3D Model: Polygonprofil P4C

Das Profil P4C hat die gleichen Eigenschaften und Funktionen wie das Profil P3G. Es ist nach DIN 32712 genormt. Ein Beispiel für die Bezeichnung: „DIN 32712 – A P4C 35“.

Parameter für den Wellen-Konfigurator:

- d_1 : Durchmesser 1
- d_2 : Durchmesser 2
- L: Länge des Absatzes

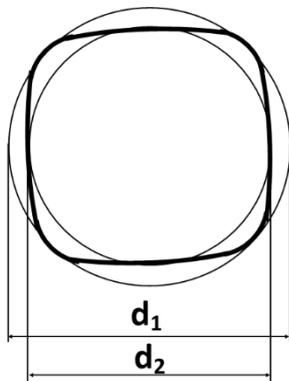


Abbildung 3.13: Skizze: Polygonprofil P4C

3.2 Weitere Elemente an Wellen

3.2.1 Zentrierbohrung

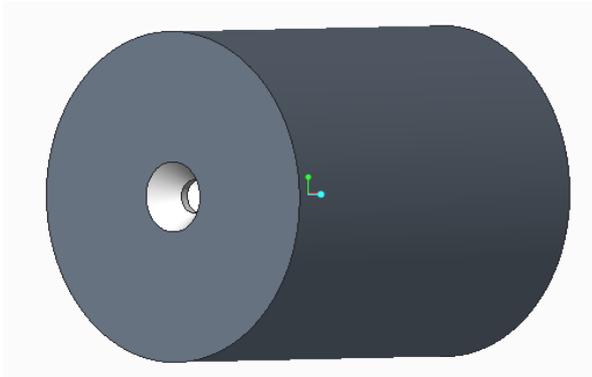


Abbildung 3.14: 3D Modell: Zylinder mit Zentrierbohrung

Bei der Gestaltung von Wellen muss die Einspannung des Werkstücks in der Drehmaschine berücksichtigt werden. Daher wird auf jeder Seite eine sog. Zentrierbohrung benötigt. Die Zentrierbohrung ist nach DIN 332-1 genormt. Für den Wellen-Konfigurator wird nur die Form A genommen. Die Zentrierbohrung wird in der Zeichnung nach DIN ISO 6411 vereinfacht dargestellt. Die Bezeichnung setzt sich zusammen aus:

1. der Normnummer
2. der Form
3. dem Durchmesser der Bohrung
4. dem Durchmesser der Senkung

Ein Beispiel für die Bezeichnung: „DIN ISO – A 2/4,25“

Parameter für den Wellen-Konfigurator:

- d: Durchmesser der Bohrung
- D_1 : Durchmesser der Senkung
- t: Tiefe

3.2.2 Radialbohrung



Abbildung 3.15: 3D Modell: Zylinder mit Radialbohrung

Eine Radialbohrung wird benötigt, falls eine Welle und eine Nabe mit einem Stift formschlüssig verbunden werden. Die Maße für diese Bohrung sollen dem entsprechenden Stift gewählt werden müssen.

Parameter für den Wellen-Konfigurator:

- D: Durchmesser
- A: Abstand von der linken Seite
- T: Tiefe

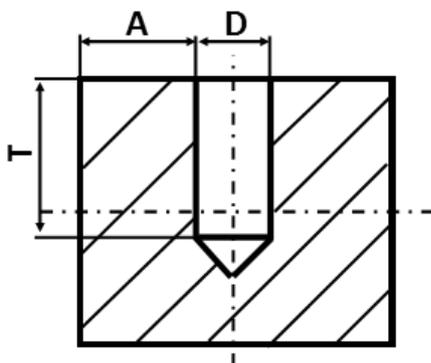


Abbildung 3.16: Skizze: Radialbohrung

3.2.3 Fase

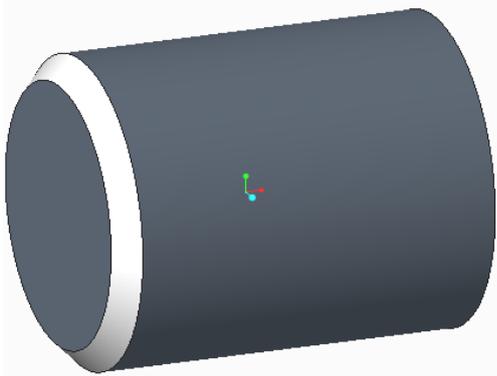


Abbildung 3.17: 3D Modell: Zylinder mit Fase

Durch die Fase an einer Kante des Absatzes werden in erster Linie die Grate entfernt, was die Verletzungsgefahr an einer scharfen Kante reduziert. Die Fase erleichtert auch das Einführen in andere Werkstücke, beispielsweise Lager.

Bei dem Modellieren und bei der Bemaßung werden der Winkel und die Fasenbreite angegeben.

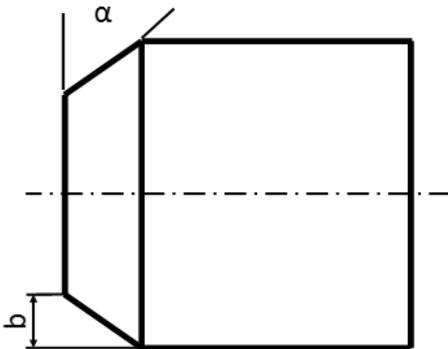


Abbildung 3.18: Skizze: Fase

α : Winkel

b : Fasenbreite

3.2.4 Rundung

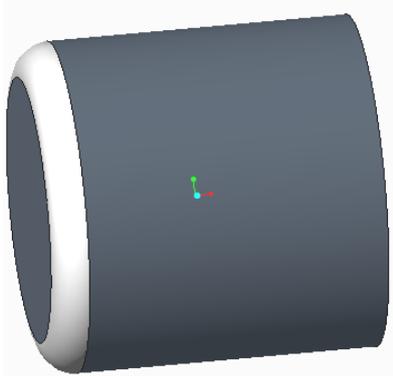


Abbildung 3.19: 3D Modell: Zylinder mit Rundung

Das Abrunden einer Kante schützt vor Verletzungen und erleichtert die Montage einer Welle in andere Bauteile. Das Abrunden einer Kante die zwischen zwei Absätzen (beim Durchmesserübergang) liegt, ist leider in einer Baugruppe nicht möglich.

Bei dem Modellieren und bei der Bemaßung wird nur der Radius der Rundung benötigt.

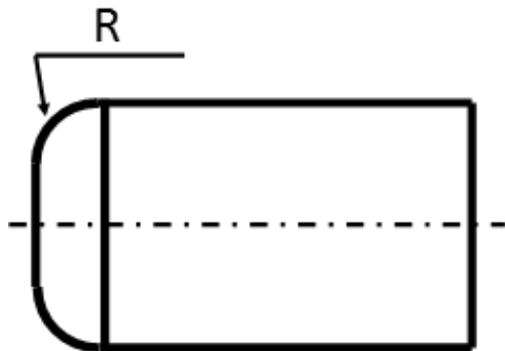


Abbildung 3.20: Skizze: Rundung

R: Radius

3.2.5 Gewinde

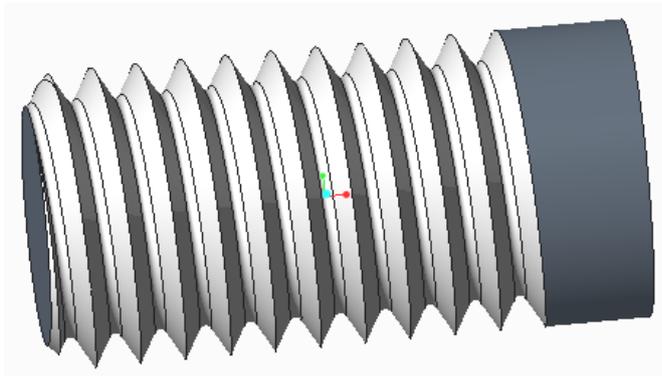


Abbildung 3.21: 3D Modell: Zylinder mit Gewinde

Ein Absatz wird mit einem Gewinde versehen, falls da z.B. ein Lager durch ein Sicherungsblech und eine Mutter gegen die axiale Verschiebung gesichert wird. Mit dem Wellen-Konfigurator kann nur das metrische Gewinde erzeugt werden. Die Abmessungen sind nach DIN 13 genormt.

In der Regel wird ein Gewinde in CAD-System als eine Abbildung auf der Mantelfläche dargestellt und nicht wie das reale(geschnittene) Gewinde. Das liegt daran, dass Präprozessoren von Gewindeschneidemaschinen es nur in dieser Form kennen und erkennen. Aus diesem Grund werden beide Formen des Gewindes in ein Modell eingefügt, das kosmetische bleibt erstmal unterdrückt. Die zusätzliche Funktion im Wellen-Konfigurator erlaubt den Wechsel der Gewindedarstellungen.

Parameter für den Wellen-Konfigurator:

- D: die Gewindebezeichnung
- L: die Gewindelänge

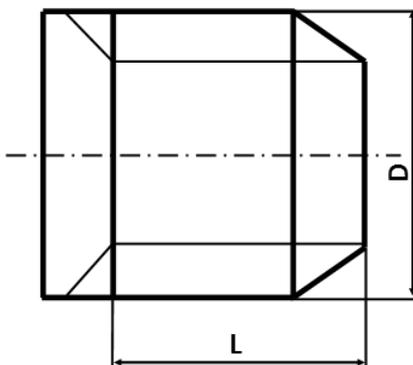


Abbildung 3.22: Skizze: Gewinde

3.2.6 Gewindefreistich

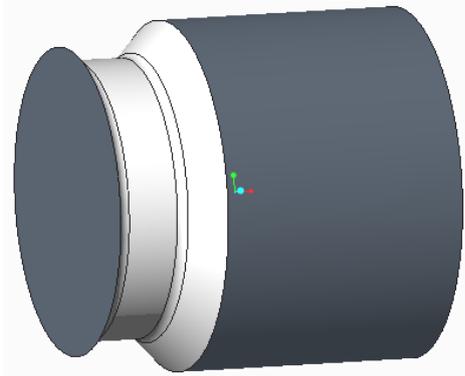


Abbildung 3.23: 3D Modell: Zylinder mit Gewindefreistich

Der Gewindefreistich wird überall dort benötigt, wo beispielsweise ein Absatz mit einem Gewinde bis zum Anschlag eingeschraubt werden soll. Die Freistiche für metrische Außengewinde sind nach DIN 76 genormt.

Parameter für den Wellen-Konfigurator:

- g_2 : Länge
- r : Radien
- d_g : Durchmesser des Gewindefreistiches

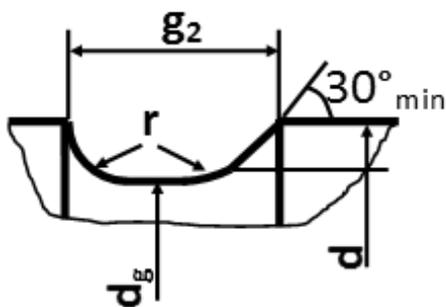


Abbildung 3.24: Skizze: Gewindefreistich

3.2.7 Freistich

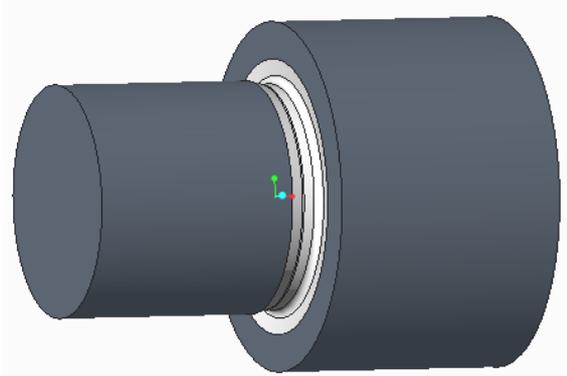


Abbildung 3.25: 3D Modell: Zylinder mit Freistich

Ein Freistich wird beispielsweise an dem Übergang zweier Absätze benötigt. Zum einen bietet er einen notwendigen Auslauf dem Bearbeitungswerkzeug. Zum anderen können so andere Bauteile wie z.B. Lager bis zum Anschlag an den anderen Absatz aufgeschoben und dagegen verspannt werden. Zuletzt reduziert er die Kerbwirkung an dem Durchmesserübergang, der als eine kritische Stelle gilt.

Die Form und die Maße eines Freistiches sind nach DIN 509 festgelegt. In dieser Arbeit werden zwei Formen (Form E und F) eines Freistichs genommen.

Beim Wellen-Konfigurator soll angegeben werden:

- die Form
- f_1 : die Länge
- r : Radien
- t_1 : Tiefe 1
- t_2 : Tiefe 2 (bei Form E kann das Feld mit der Tiefe 2 frei gelassen werden)

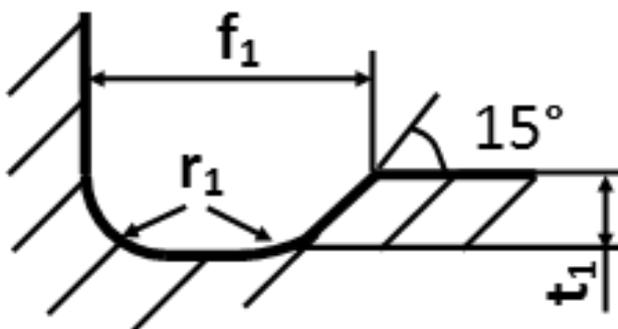


Abbildung 3.26: Skizze: Freistich Form E

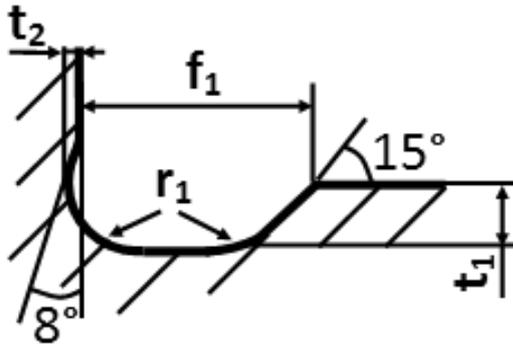


Abbildung 3.27: Skizze: Freistich Form F

3.2.8 Passfedernut Form A

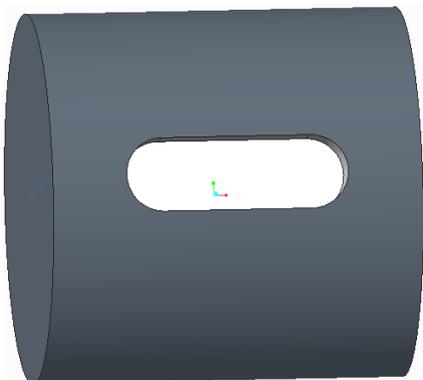


Abbildung 3.28: 3D Modell: Zylinder mit Passfedernut Form A

Eine Passfedernut wird erzeugt, wenn eine Welle das Drehmoment mit der formschlüssigen Welle-Nabe Verbindung übertragen soll. In diese Nut wird eine Passfeder eingelegt, die zu einem Teil in der Welle und zum anderen in der Nabe liegt. Es gibt verschiedene Ausführungen von Passfedern, für den Wellen-Konfigurator wird nur eine (Form A) genommen. Diese Form ist nach DIN 6885 A genormt.

Parameter für den Wellen-Konfigurator:

- l: Länge
- b: Breite
- t_1 : Tiefe
- a: Seitenabstand
- optional kann auch die Winkellage angegeben werden

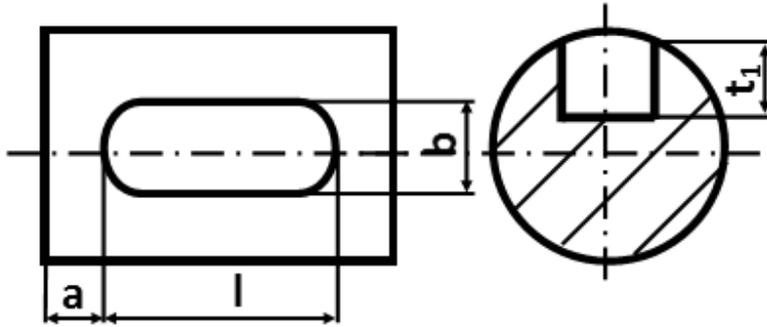


Abbildung 3.29: Skizze: Passfedernut Form A

3.2.9 Nut für Sicherungsringe

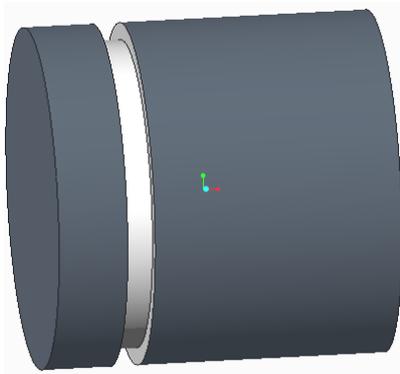


Abbildung 3.30: 3D Modell: Zylinder mit Nut für Sicherungsringe

Ein Sicherungsring verhindert die axiale Verschiebung von Bauteilen auf einer Welle. Um diese Bauteile fixieren zu können, benötigt dieser Ring eine Nut. Es gibt unterschiedliche Arten dieser Ringe, aber die Form der Nut bleibt bei allen gleich. Daher sollen die Maße der Nut den entsprechenden Ringen angepasst werden.

Parameter für den Wellen-Konfigurator:

- n : Abstand von der linken Seite des Absatzes
- m : Nutbreite
- d_2 : Durchmesser

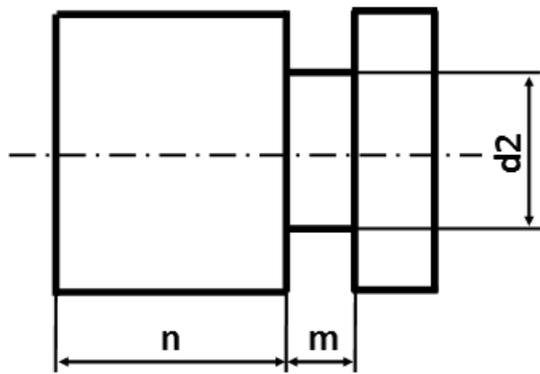


Abbildung 3.31: Skizze: Nut für Sicherungsringe

3.2.10 Nut für Sicherungsblech

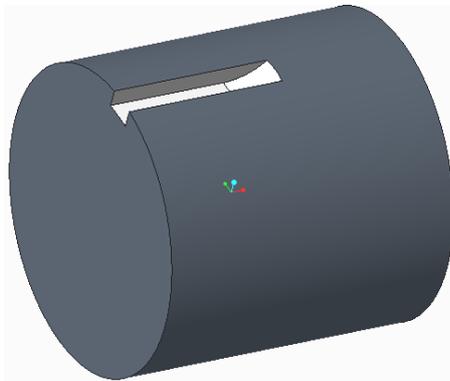


Abbildung 3.32: 3D Modell: Zylinder mit Nut für Sicherungsblech

Eine andere Möglichkeit, Bauteile auf einer Welle gegen axiale Verschiebung fixieren zu können, ist die Verwendung von Muttern. Diese Muttern werden zusammen mit sog. Sicherungsblechen verwendet. Das Sicherungsblech soll die Mutter gegen das Losdrehen sichern. Aus diesem Grund muss es selbst auf der Welle gesichert werden. Dafür kommt eine Nut zum Einsatz. Die Nut wird genau wie das Sicherungsblech nach DIN 5406 bemaßt.

Parameter für den Wellen-Konfigurator:

- L: Länge
- B: Nutbreite
- T: Tiefe
- R: Radius des Auslaufs

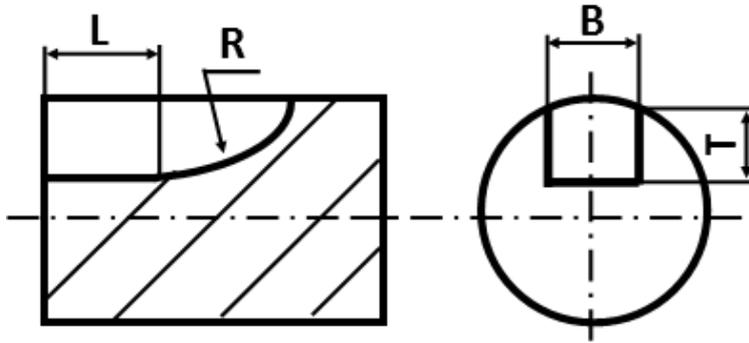


Abbildung 3.33: Skizze: Nut für Sicherungsblech

3.3 Schwierigkeiten

Bei der Modellierung der Schablonen sind folgende Schwierigkeiten aufgetreten:

eine davon wird am Beispiel eines Vielnutprofils DIN ISO 14 (sog. Keilwelle) genauer erläutert:

- Die Konstruktion und die Fertigung einer Welle sind genormt. D.h. um ein Keilwellenprofil zu modellieren bräuchte man die jeweilige Norm. Da aber die Studierenden keine der Normen besitzen und nur geringe Möglichkeit, in Form von kleinen Auszügen in Lehrbüchern (wie z.B. Tabellenbuch Metall oder Roloff /Matek Maschinenelemente) haben darein zu sehen, ist die hundertprozentige normgerechte Modellierung praktisch nicht möglich. Die Auszüge liefern die Informationen nicht über alle möglichen Keilwellenprofile und nicht über alle Bemaßungen. Man findet dort die benötigten Durchmesser, die Zahnbreite etc., aber nichts darüber wie groß die Fase an einem Ende oder der Durchmesser des Werkzeugauslaufs sein sollen [vgl. Ro/Ma 15]

Diese Informationen sind nicht zwingend erforderlich, denn bei der Zeichnung werden die Keilwellenprofile nicht vollständig bemaßt, sondern es werden nur deren Kurzbezeichnungen (z.B. DIN ISO 14 – W 8x32x36) [vgl. T. Z. 3 A.] angegeben. Für die Modellierung kann man also eigene plausible Werte für die fehlenden Bemaßungen verwenden.
- Die andere große Schwierigkeit stellt die automatische Ableitung einer Zeichnung dar. Denn während man weiß aus welchen Elementen eine Welle bestehen könnte und dafür die benötigten Schablonen (sog. Templates) modellieren kann, ist es völlig ungewiss wie diese Schablonen zusammengesetzt werden. Daher ist es kaum möglich diese Ableitung so zu implementieren, dass eine Welle als Ganzes bemaßt wird, alle Schnitte von verborgenen Elementen (z.B. Bohrung) gemacht werden, der richtige Maßstab gewählt wird usw. Man kann natürlich versuchen z.B. einzelne Absätze zu bemaßen, dabei läuft man die Gefahr, dass die Zeichnung nicht normgerecht sein wird. Die Änderung der Zeichnung würde dann noch mühseliger und zeitintensiver sein, als die manuelle Zeichnungsableitung.

4 Implementierung der Funktionen des Wellen-Konfigurators in VB API

Um auf die Schablonen zuzugreifen und sie zu verändern, gibt es In Creo unterschiedliche Schnittstellen (C++-, Java-, VB-API). Aus Erfahrungsgründen wird in dieser Arbeit die VB Schnittstelle verwendet.

4.1 Allgemeine Informationen zu VB API

1. Die VB API (Application Programming Interface) bietet die Möglichkeit für die Anbindung einer externen Software, um eigenen Konstruktionsprozess zu automatisieren, indem man beispielsweise:

- seine Software bestimmte Berechnungen durchführen lässt und somit durch Schleifen, Fallunterscheidungen oder Regeln:
 - Parameter eines Bauteils verändert
 - Eine Baugruppe zusammenstellt
 - Mitglieder einer Familientabelle editieren
 - usw...

- Dokumentationen erstellt:
 - Datenblätter
 - Zeichnungen
 - usw...

2. Externe Applikation ist ein COM-Client und hat den Zugriff auf alle Objekte, die es in PTC Creo gibt. D.h. man nutzt in seinem Code alle Funktionen, Methoden und Eigenschaften aller Objekte.

3. Es kann nur eine Verbindung zwischen Applikation und Creo Sitzung existieren

4. Grundkenntnisse über VB-Programmierung und Konstruktion in Creo müssen vorhanden sein. Alle Einzelheiten über Creo VB API sind im mitgelieferten „VB API User’s Guide“ zu finden.

Alle für Wellen-Konfigurator verwendeten Objekte werden in folgenden Unterkapiteln noch genauer erläutert.

4.2 Programmierumgebung

Um eigene Applikation mit VB API zu implementieren stehen zwei Programmierumgebungen zur Verfügung:

1. Microsoft Visual Studio
2. Microsoft Office (z.B. Excel)

Der Wellen-Konfigurator wird in Microsoft Visual Studio 2013 programmiert und kann als eine Anwendung durch eine Verknüpfung auf dem Desktop oder durch ein Mapkey in Creo gestartet werden. (Achtung: Den Wellen-Konfigurator starten, nur nachdem eine Creo Baugruppen Sitzung gestartet ist)

Unabhängig davon ob man nur den Wellen-Konfigurator nutzen oder selbst eine Applikation entwickeln möchte, sollte man folgende Vorbereitungen treffen:

1. Damit zwischen der Applikation und der Creo Sitzung eine Verbindung hergestellt wird, sollte PRO_COMM_MSG_EXE als Systemvariable mit dem vollständigen Pfad zu pro_comm_msg.exe Ausführung-Datei gesetzt werden. Diese Datei befindet sich in der Regel in <PTC Ordner><Creo><Versionnummer><Common Files><Rechnersystem-Typ><obj>

Beispiel: „\PTC\Creo 3.0\M030\Common Files\x86e_win64\obj\pro_comm_msg.exe“

2. Die Systemvariable PRO_DIRECTORY sollte mit ihrem Pfad <PTC Ordner><Creo><Versionnummer> gesetzt werden
3. Dann sollte der COM Server registriert werden. Die Datei vb_api_register.bat, die sich unter <PTC Ordner><Creo><Versionnummer><Parametric><bin> befindet, als Administrator ausführen.
4. Sich vergewissern, dass Microsoft .NET Framework auf dem Rechner installiert ist. (Falls man Visual Studio installiert hat, ist der Schritt schon erledigt)
5. Wenn die Verbindung zwischen den Beiden besteht, dann sollten folgende Prozesse laufen:
 - pro_comm_msg.exe
 - nmsd.exe
 - pfclscom.exe
6. Wenn man selbst eine Applikation entwickelt, sollte man nicht vergessen in der Programmierumgebung den Verweis auf Creo Parametric Bibliothek hinzuzufügen.

4.3 Applikationshierarchie

An diesem Beispiel wird die Reihenfolge, wie eine Applikation aufgebaut werden soll, erklärt. Konkret wird es hier nach allen Achsen von einer Bohrung gesucht.

1. Im Header muss *Imports pfcls* angegeben werden
2. Verbindung zu PTC Creo herstellen:
Dim connection As IpfcAsyncConnection
Dim classAsyncConnection As New CCpfcAsyncConnection

```
connection = classAsyncConnection.Connect (DBNull.Value, DBNull.Value,  
DBNull.Value, DBNull.Value)
```

3. Aktuelle Sitzung ansprechen:
Dim session As IpfcBaseSession
session = connection.Session
4. Alle Modelle in der Sitzung auflisten:
Dim models As IpfcModels
models = session.ListModels()
5. Das Modell, in dem sich die Bohrung befinden soll, ansprechen:
Dim model As IpfcModel
model = models[0]
6. Alle Features, die das Modell hat, auflisten:
Dim items As IpfcModelItems
items = model.ListItems (EpfcModelItemType.EpfcITEM_FEATURE)
7. Alle Features nach Typ Bohrung absuchen:
if (feature.FeatType = EpfcFeatureType.EpfcFEATTYPE_HOLE) then
8. Alle Achsen, die die Bohrung besitzt, auflisten:
Dim axes As IpfcModelItems
axes = feature.ListSubItems (EpfcModelItemType.EpfcITEM_AXIS)

4.4 Funktionen des Wellen-Konfigurators

In diesem Abschnitt geht es darum, die Funktionen des Wellen-Konfigurators zu präsentieren. Dabei liegt der Schwerpunkt auf der Erklärung der verwendeten API Objekten, deren Funktionen, Methoden und Eigenschaften.

Der Wellen-Konfigurator wird modular aufgebaut. D.h. er wird aus kleinen Teilfunktionen, die alleine oder in Zusammenarbeit jeweils eine Aufgabe erfüllen, aufgebaut. Das soll die Übersichtlichkeit und die Verständlichkeit der ganzen Applikation erleichtern. Gleichzeitig soll dadurch die Flexibilität der Applikation gewährleistet werden.

Übersicht der Funktionen bzw. Prozeduren:

1. Verbindung herstellen
2. Verbindung trennen
3. Wellenabsatz löschen
4. Parameter ändern
5. Aufruf eines Modells
6. Aufruf einer Schablone

7. Eine Komponente in Baugruppe einfügen
8. Modelle in der Baugruppe umordnen
9. Alle Modelle in der Baugruppe auflisten
10. Alle Modelle in einer Liste darstellen
11. Ein Model aktualisieren (regenerieren)
12. Ein Feature eines Modells unterdrücken
13. Ein Element löschen
14. Ein Feature wieder sichtbar machen
15. UDF in ein Modell einfügen
16. Zeichnung ableiten

Zur Lösung des Gewinde-Problems [3.2.5] wird zusätzlich eine Funktion entwickelt

17. Gewindedarstellungen wechseln

4.4.1 Verbindung herstellen

Die Verbindung zwischen einer externen Applikation und einer Creo Sitzung ermöglicht die Klasse `IpfcAsyncConnection`. Wenn man eine neue Verbindung herstellen möchte, dann nutzt man die Funktion `connect()`, deren vier Parameter auf null gesetzt werden können.

```
connection = New CCpfcAsyncConnection.Connect(DBNull.Value, DBNull.Value,  
DBNull.Value, DBNull.Value)
```

Dann soll die aktuelle Sitzung angesprochen werden. Die Klasse `IpfcBaseSession` enthält alle Informationen über jede Creo Parametric Sitzung.

```
session = connection.Session
```

4.4.2 Verbindung trennen

Zur Trennung der Verbindung dient die Funktion `disconnect()` aus der Klasse `IpfcAsyncConnection`, mit einem Zeitparameter, der beispielsweise auf 1 gesetzt werden kann.

```
connection.Disconnect(1)
```

4.4.3 Wellenabsatz löschen

Die Aufgabe dieser Funktion besteht hauptsächlich darin, ein Modell aus einer Baugruppe zu löschen. Die große Herausforderung dabei ist die Stabilität der Baugruppe nach dem Entfernen des ausgewählten Modells. Das Entfernen eines Modells lässt sich in drei Fälle unterscheiden:

1. Das Erste Modell (der erste Absatz der Welle) der Baugruppe wird entfernt: D.h. das nächste Modell soll die Position vom ersten einnehmen und alle anderen Modelle sollen ihm folgen.
2. Das letzte Modell kann ruhig gelöscht werden, hat also keine Auswirkungen auf die Stabilität
3. Ein Model, das sich zwischen zwei anderen befindet. In diesem Fall soll das nachfolgende Modell alle Nebenbedingungen (Constraints) vom gelöschten bekommen und dessen Position einnehmen. Damit wird sichergestellt, dass der Nachfolger mit dem Vorgänger zusammengeführt werden und die Welle wieder ganz wird.

Die Funktion besteht aus zwei Teilen. Ein Teil gehört zum Hauptprogramm, das die Verbindung herstellt. Der andere Teil gehört zu einer Windows Form und somit dient der Bedienung.

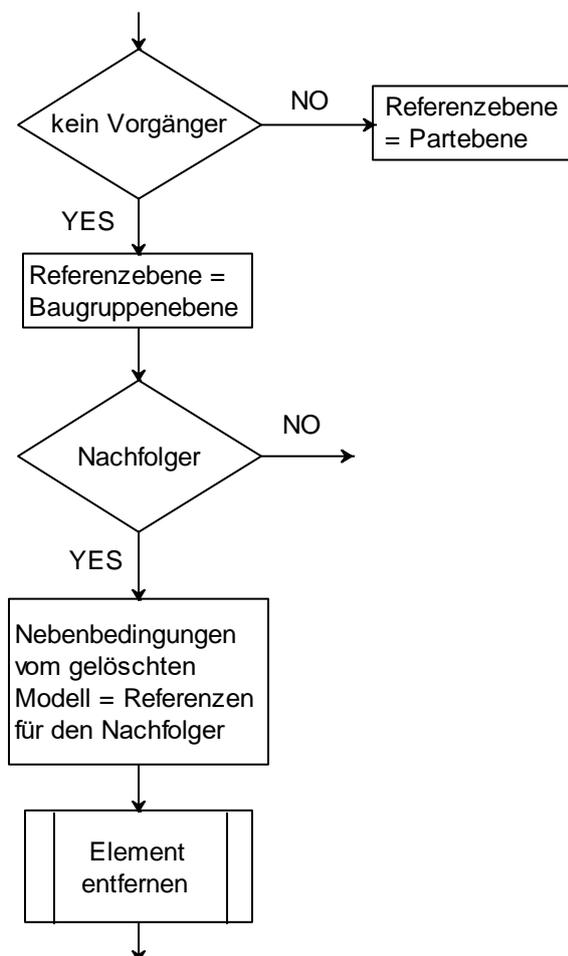


Abbildung 4.1: Flussdiagramm: Komponente löschen

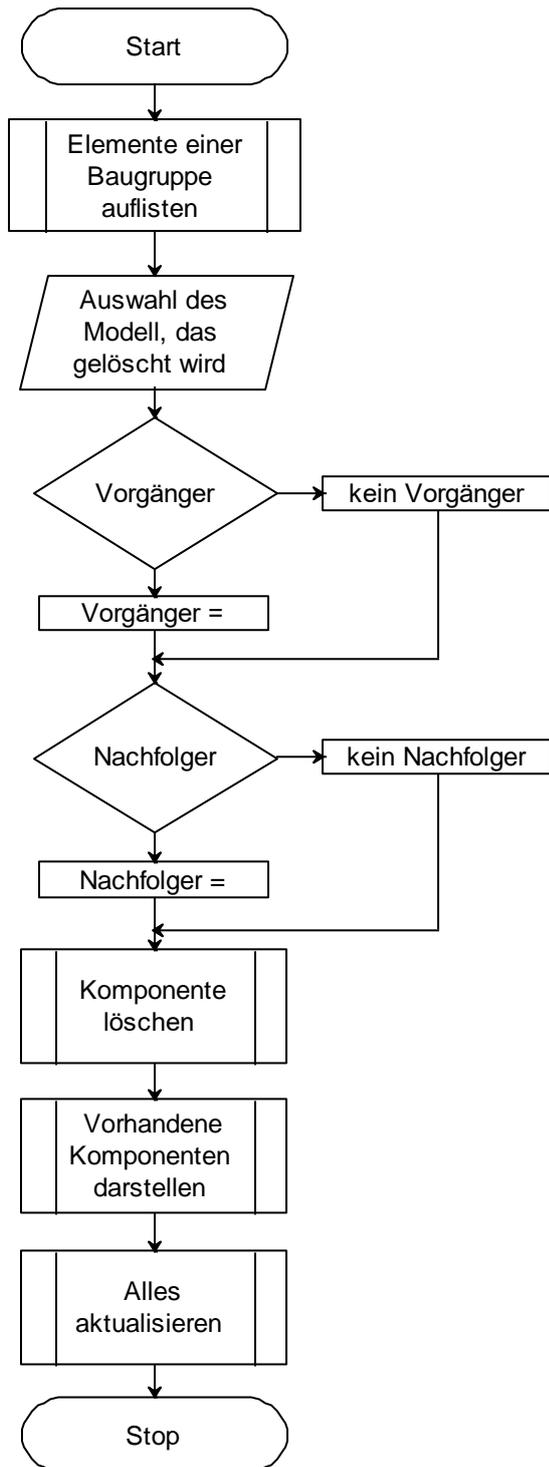


Abbildung 4.2: Flussdiagramm: Wellenabsatz löschen

Bei dieser Funktion verwendeten Klassen:

1. IpfcSolid: diese Klasse definiert einen Festkörper und führt die angegebenen Operationen aus, in diesem Fall ist es das Löschen einer Komponente. Wenn es sich um eine Baugruppe handelt, dann ist der Festkörper die Baugruppe selbst und die einzelnen Modelle sind die Komponenten.

Wenn es sich um einen Part handelt, dann ist das Modell das Solid und dessen Elemente (Ebenen, Achsen, UDFs...) sind die Komponenten.
Sub ExecuteFeatureOps()

2. IpfcFeature: diese Klasse definiert eine Komponente und erzeugt eine gewünschte Operation (Löschen)
Function CreateDeleteOp()
3. IpfcAssembly: Diese Klasse beschreibt eine Baugruppe. Eine Variable dieses Typs wird der aktuellen Sitzung gleichgesetzt.
4. IpfcComponentPath: Diese Klasse enthält die Informationen über die Pfade der einzelnen Komponenten in der Baugruppe. Diese Informationen werden benötigt, um Referenzen auf die gewünschten Komponenten zu setzen.
5. IpfcModelItem: Diese Klasse beschreibt eine Komponente eines Modells. In dieser Funktion sind es die Ebenen, die ausgewählt werden und als Referenzen dienen sollen.
6. IpfcModel: Diese Klasse enthält alle Informationen über ein Modell, das sich aktuell in der Sitzung befindet.
7. IpfcModelItemOwner: Diese Klasse wird benötigt, um ein Objekt für die IpfcModelItem Klasse durch die Funktion GetItemById oder GetItemByName zu erzeugen.
8. IpfcSelection: Diese Klasse repräsentiert ein ausgewähltes Objekt mit seinen Eigenschaften.
9. IpfcComponentConstraint: Diese Klasse repräsentiert Nebenbedingungen eines Objekts. Z.B. in diesem Objekt wird es festgelegt, wie zwei Ebenen zueinanderstehen sollen: zusammenfallend, in einem Abstand usw.
10. Cintseq: Mit dieser Klasse kann man ein Objekt erstellen, das die ID Nummer einer Komponente enthält. Dieses Objekt wird bei der Angabe des Pfades zu einer Komponente benötigt.

4.4.4 Parameter ändern

Parameter_ändern ist eine kleine Prozedur, die die vordefinierten Parameter eines Modells aufruft und deren Werte verändert.

Die Methode CreateDoubleParamValue() der Klasse CMpfcModelItem erzeugt ein Objekt mit dem neuen Wert, der einer Variable der Klasse IpfcParamValue zugewiesen wird. Danach soll

einer Variable der Klasse `IpfcParameterOwner` ein Modell, das den Parameter enthält, übergeben. Die Methode `GetParam()` der Klasse `IpfcParameter` greift auf den Parameter im Modell zu und die Methode `SetScaledValue()` setzt den neuen Wert.

4.4.5 Aufruf eines Modells

Mit dieser Funktion können nur die Modelle aufgerufen werden, die aktuell in der Sitzung sind. Die Methode `CreateFromFileName()` der Klasse `IpfcModelDescriptor` erzeugt ein Objekt mit einem angegebenen Namen. Die Methode `RetrieveModel()` der Klasse `IpfcBaseSession` ruft das Modell auf. Dafür benutzt sie das `ModelDescriptor`-Objekt als Parameter.

4.4.6 Aufruf einer Schablone

Der Aufruf einer Schablone funktioniert ähnlich wie der Aufruf eines Modells. Der Unterschied zwischen den beiden besteht darin, dass die Schablone sich nicht in aktueller Sitzung befindet. Sie kann an einem beliebigen Ort sein und von dort aufgerufen werden. Dafür muss man dem Objekt der Klasse `IpfcModelDescriptor` einen Pfad zuweisen und ihm sagen um welchen Typ (Assembly oder Part) es sich bei dem angegebenen Modell handelt. Hier benutzt man dann nicht die Methode `RetrieveModel()`, sondern die `RetrieveModelWithOpts()`.

4.4.7 Eine Komponente in Baugruppe einfügen

Diese Prozedur ist eine der umfangreichsten und kompliziertesten Funktionen. Erstens muss die Position der Komponente volldefiniert werden. Dafür werden die drei Ursprungebenen der Baugruppe mit den der Komponente mit der Bedingung Abstand in Verhältnis gesetzt. Damit kann man beim Bedarf die Komponente in z- oder y-Richtung verschieben, falls die Welle nicht konzentrisch sein soll. Dafür dienen die Vorderansicht- und Draufsichtsebene als Referenzen. Für die x-Richtung müssen die Seitenebenen als Referenzen gewählt werden, die immer zusammenfallend sind. Da gibt es zwei Fälle zu unterscheiden:

1. Komponente wird in eine leere Baugruppe eingefügt. Dafür muss die Seitenebene der Baugruppe als Referenz gewählt werden.
2. Die Seitenebene einer Komponente dient als Referenz

In der Regel werden alle Komponenten in die Baugruppe eingebaut. Falls man aber möchte, dass die neue Komponente zwischen zwei vorhandenen eingebaut wird, dann muss man den Modellbaum umordnen. Dazu dient die Prozedur die Modelle in der Baugruppe umordnen, die später genau erläutert wird. Dabei werden die Bedingungen zwischen den vorhandenen Komponenten aufgelöst, dann wird die neue eingefügt, und zum Schluss müssen neue Bedingungen zwischen den dreien definiert werden.

Zum Einfügen muss man immer angeben, hinter welche Komponente die neue eingefügt werden soll.

Diese Funktion nutzt die gleichen Klassen und Methoden wie die Wellenabsatz löschen. Eine Ausnahme stellt die Methode AssembleComponent() der Klasse IpfcAssembly dar, die eine Komponente in Baugruppe einbaut.

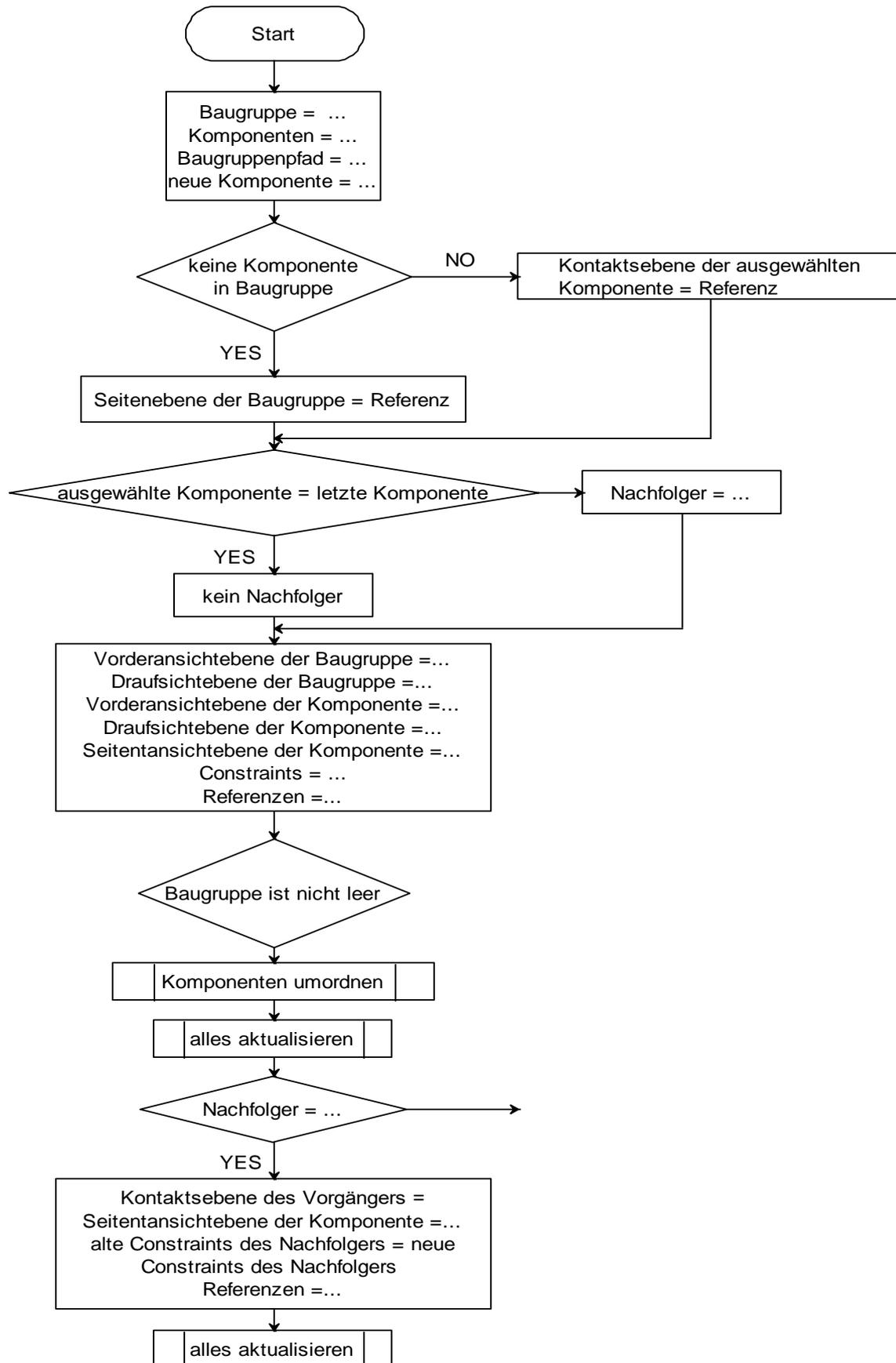


Abbildung 4.3: Flussdiagramm: eine Komponente in Baugruppe einfügen

4.4.8 Modelle in einer Baugruppe umordnen

Wie es schon erwähnt wurde, werden Komponenten nacheinander in eine Baugruppe eingebaut. In der Reihenfolge erscheinen sie auch im Modellbaum. Daraus folgt, man kann eine Komponente (K1) als Referenz für eine andere (K2) nur dann nutzen, wenn die K1 vor K2 im Baum steht. Man kann es so ausdrücken, für ein Modell im Baum sind nur seine Vorgänger sichtbar, aber nicht seine Nachfolger. D. h. also wenn ein neues Modell eingebaut wird, dann ist es im ersten Moment für alle anderen unsichtbar, und man kann kein vorhandenes Modell hinter ihm setzen. Genau das wird aber gefordert, wenn ein neuer Wellenabsatz zwischen zwei Absätzen „eingeschoben“ werden soll. Aus diesem Grund soll der Absatz zuerst auf die gewünschte Position im Baum gesetzt werden und dann erst in die Baugruppe. Dazu dient die Prozedur `reorderaftercomp()`, die eine neue Komponente hinter die ausgewählte Komponente setzt und somit sichtbar für deren Nachfolger macht.

Bei dieser Funktion verwendeten Klassen:

1. `IpfcFeatureOperation` definiert eine Komponentenoperation.
2. `IpfcReorderAfterOperation` definiert, dass eine Komponente hinter eine andere gesetzt werden soll. Hier wird ein neues Objekt mit der Methode `CreateReorderAfterOp()` erzeugt. Das Objekt hat die Komponente als Parameter und wird zu `FeatureOperations` zugefügt. Die erstellte Operation wird mit der Methode `ExecuteFeatureOps()` ausgeführt.

4.4.9 Alle Modelle in der Baugruppe auflisten

Diese Funktion liefert alle sichtbaren Modelle, die in der Baugruppe sind. Sie wird praktisch in jeder Form des Wellen-Konfigurators benötigt. Alle Modelle werden in einer Liste dargestellt und können, beispielsweise um ein Modell zu löschen, ausgewählt werden.

Bei dieser Funktion verwendeten Klassen:

1. `IpfcAssembly`: Der Variable dieses Typs wird der Wert der aktuellen Baugruppe zugewiesen.
2. `IpfcFeatures`: Hier werden alle Komponenten mit der Methode `ListFeaturesByType()` gespeichert. Die Methode hat folgende Parameter:
 1. `VisibleOnly` as Boolean: Hier wird bestimmt, ob die Modell sichtbar sind oder nicht (True or False)
 2. `Type` as `IpfcFeatureType`: Hier wird der Typ der Komponente angegeben. In diesem Fall ist es `EpfcFeatureType.EpfcFEATTYPE_COMPONENT`

4.4.10 Alle Modelle in einer Liste darstellen

Diese Prozedur ruft erstmal die Funktion „Modelle in der Baugruppe auflisten“ auf. Eine For-Schleife durchläuft alle Komponenten und speichert in einem Feld deren Namen. Die Eigenschaft ModelDescr liefert mit der Funktion GetFullName() den vollen Namen einer Komponente. Die Namen werden dann in einer ListBox dargestellt.

4.4.11 Ein Modell aktualisieren

Diese Prozedur besteht aus zwei Prozeduren der Creo VB API.

1. Regenerate (): aktualisiert den Festkörper (Assembly/Part)
2. Rerfresh(): aktualisiert das aktive Fenster

Bei dieser Funktion verwendeten Klassen:

1. IpfcSolid [Kap. 4.4.3]
2. IpfcWindow: Diese Klasse beschreibt die Attributen eines Fensters

4.4.12 Ein Feature eines Modells unterdrücken

Hier sollen zunächst ein Modell als Solid und ein Element als Feature definiert werden. Danach wird ein Objekt der Unterdrücken-Operation CreateSuppressOp erzeugt, das von der Methode ExecuteFeatureOps ausgeführt wird. Die SuppressOP gehört der Klasse IpfcSuppressOperation. Mit der Methode AllowGroupMembers = True erlaubt man das Unterdrücken ganzer Gruppe von Features (UDFs). D. h. mehrere Features können zu einer Gruppe zusammengefasst und mit einem Befehl unterdrückt oder gelöscht werden.

4.4.13 Ein Feature löschen

Diese Prozedur ist die Unterprozedur der Funktion Wellenabsatz löschen. Sie funktioniert ähnlich wie die Prozedur ein Feature unterdrücken. In diesem Fall soll nur ein Objekt mit der Entfernen-Operation erzeugt werden. Die DeleteOp gehört der Klasse IpfcDeleteOperation.

4.4.14 Ein Feature wieder sichtbar machen

Diese Prozedur ist ein Mitglied der Feature-Operationen (Unterdrücken-Operation, Entfernen-Operation). Sie funktioniert absolut gleich wie die beiden vorherigen Prozeduren. Hier wird nur ein Objekt mit der Wiederaufnahme-Operation erzeugt. Die ResumeOp gehört der Klasse IpfcResumeOperation.

4.4.15 UDF in ein Modell einfügen

Weitere Elemente einer Welle werden als UDFs eingefügt. Jedes UDF braucht eine eigene Funktion, weil alle UDFs unterschiedliche Anzahl von Referenzen und Dimensionen haben. Die Funktion erstellt eine Kopie von vorhandener UDF-Schablone und platziert sie in einem Modell. Das Modell, die Referenzen (Kanten, Ebenen, ...) und die Dimensionen (Länge, Breite, ...) werden als Parameter übergeben. Zuerst soll ein neues Objekt mit sog. Anweisungen erzeugt werden. Diese Anweisungen beinhalten Referenzen, Dimensionen und Richtungen eines UDFs. Es müssen also alle Referenzen ausgewählt werden und in einem Objekt gesammelt werden. Dann werden alle Dimensionen definiert und auch in einem Objekt gesammelt. Optional kann man die Richtungen der Schnitte bestimmen. Beispielsweise wird ein Gewinde in der Regel von der Seitenansichtsebene geschnitten. Falls es aber von der anderen Seite geschnitten werden sollte, muss man die Richtung des Schnittes ändern. Alle diese gesammelten Objekte werden der Kopie des UDFs übergeben.

Bei dieser Funktion verwendeten Klassen:

1. IpfcUDFCustomCreateInstructions. Die Methode create(Name des UDFs) erzeugt eine Kopie von UDF.
2. IpfcSelection: [Kap 4.4.3]
3. IpfcUDFReference: Diese Klasse beschreibt ein Feld mit Referenzelementen.
4. IpfcUDFVariantDimension: Diese Klasse gibt den Wert einer Dimension an.
5. IpfcUDFVariantValues: Diese Klasse beschreibt ein Feld mit allen Dimensionen eines UDFs
6. IpfcFeatureGroup: Diese Klasse beschreibt eine Feature-Gruppe. Die Methode CreateUDFGroup(UDFInstructions) platziert ein UDF.
7. IpfcUDFOrientations: Diese Klasse beschreibt ein Feld, das die Informationen besitzt, ob Richtungen eines UDFs geändert werden sollen.

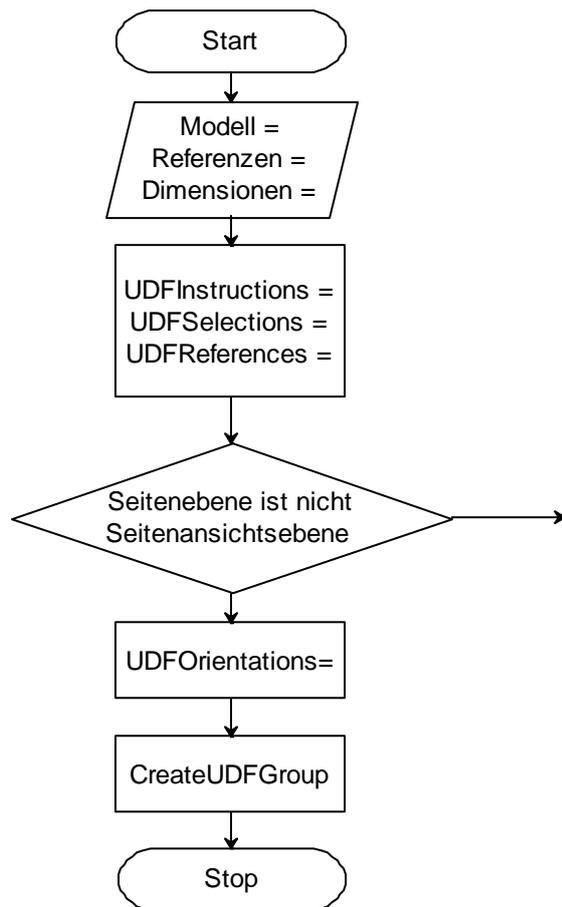


Abbildung 4.4: Flussdiagramm: UDF einfügen

4.4.16 Zeichnung

Die Problematik der Ableitung einer Zeichnung wurde schon erwähnt. Dennoch wird es mit dieser Funktion versucht teilweise eine Zeichnung abzuleiten. Diese Prozedur öffnet zuerst ein neues Zeichnung-Fenster. Danach wird eine Vorderansicht von einer Welle dargestellt. Zum Schluss werden die Längen und die Durchmesser der einzelnen Absätze bemaßt. Um eine Bemaßung der Länge zu erzeugen, werden die beiden Seitenebenen ausgewählt. Um die Durchmesser zu erzeugen werden zusätzliche zwei Punkte in Modellen eingefügt. Der Abstand zwischen diesen Punkten stellt auch in einer Zeichnung einen Durchmesser dar.

Die wichtigsten Klassen dieser Funktion sind:

1. `IpfcDrawing`: Diese Klasse beschreibt eine Zeichnung. Die Methode `CreateDrawingFromTemplate()` erzeugt eine Zeichnung. Als Parameter für diese Methode dienen der Name der Zeichnung, die gewünschte Schablone des Blattes und das Modell, das dargestellt werden soll.
2. `IpfcView2D`: Diese Klasse repräsentiert eine Ansicht. Die Methode `CreateView()` erzeugt die Ansicht.
3. `IpfcDrawingDimCreateInstructions`: In diesem Objekt werden alle Anweisungen für die Erzeugung einer Bemaßung gesammelt.

4. IpfcDrawingDimensionShowInstructions: Hier wird bestimmt in welcher Ansicht die Bemaßung erzeugt werden soll

Da diese Funktion nicht ihren ganzen Zweck erfüllt, wird sie wahrscheinlich kaum benutzt. Daher wird sie nicht weiter behandelt.

4.4.17 Gewindedarstellungen wechseln

Diese Funktion durchläuft alle Modelle und sucht nach Gewinde-Features. Sobald sie ein geschnittenes Gewinde entdeckt unterdrückt sie dieses. Danach wird das kosmetische Gewinde wieder sichtbar gemacht. Wie man Features unterdrückt und wieder sichtbar macht wurde schon erklärt.

5 Dialogorientierte Benutzeroberfläche

In diesem Kapitel werden alle Bedienelemente des Wellen-Konfigurators vorgestellt.

Jedes Element, sei es ein Modell oder ein Feature, aus der Produktarchitektur braucht ein eigenes Fenster, wo man sehen kann:

- welche Angaben werden gefordert, um das jeweilige Element zu erzeugen
- eine Skizze mit Legende, die die Angaben zeigt
- in welches Modell soll ein Feature eingebaut werden
- einen Knopf zum Einfügen des Elements
- einen Knopf zum Schließen des Fensters

Dafür wird in Visual Studio für jedes Element eine Windows Form angelegt. Das soll für die Übersichtlichkeit der Funktionalitäten und Benutzerfreundlichkeit sorgen.

Nachdem man den Wellen-Konfigurator gestartet hat, erscheint die Haupt-Form. In dieser Form kann man auswählen:

- den Schnitttyp eines Wellenabsatzes
- weitere Elemente einer Welle
- einen Wellenabsatz löschen
- unvollständige Zeichnung erstellen
- mit dem Knopf Schließen schließt man die ganze Applikation und trennt die Verbindung zu Creo

Durch die Auswahl einer Funktion oder eines Elements erscheint die dazugehörige Form.

Für die Knöpfe gibt es in Visual Studio das Steuerelement Button, das ein Ereignis auslöst, wenn man darauf klickt.

Alle Wellenabsätze, die schon in der Baugruppe sind, werden in einer ListBox dargestellt. Aus dieser Liste wählt man aus, welcher Wellenabsatz gelöscht oder hinter welchen ein neuer eingefügt werden soll.

Zur Darstellung von Namen der Eingabewerten dient das Steuerelement Label.

Neben den Namen steht immer eine TextBox. Dieses Steuerelement ermöglicht die Eingabe von Texten (Namen für Wellenabsätze) oder Zahlen (Parameterwerte).

Die Skizzen werden in PictureBoxen angezeigt.

Das Steuerelement ComboBox zeigt ein Textfeld mit einer Liste zulässiger Werte an. In so einer ComboBox werden die Namen einzelner Wellenelemente gespeichert und können ausgewählt werden. Die Änderung des Wertes dieser Box löst das Erscheinen einer Windows Form aus.

Zu den Eingabewerten gehören:

- Parameter eines Modells oder UDFs
- Seite als Referenz zum Platzieren

- Verschiebungen in z- und y-Richtung, falls die Welle nicht konzentrisch sein soll
- Benennung eines Absatzes

6 Fazit und Ausblick

Das Ziel dieser Arbeit war die Entwicklung eines Wellen-Konfigurator, der in der Lage ist auf Basis einiger Anforderungs- und Auslegungskriterien eine mechanische Welle im CAD-System vollautomatisch zu generieren. Das gesetzte Ziel wurde auch erreicht. Es ist möglich eine externe Software, die die benötigten Funktionalitäten beinhaltet, in Form eines Plugins in PTC Creo zu integrieren. Dadurch wurden viele Erfahrungen in einem völlig neuen und unbekanntem Bereich der Konstruktion gewonnen. Dabei wurden die ersten Lücken in eigenen Fähigkeiten aber auch in Lehrinhalten des Maschinenbaus entdeckt. Bei dieser Arbeit sollte an sich kein neues Produkt erfunden werden. Die Marktanalyse ergab, dass schon viele Unternehmen sich mit dem Thema beschäftigt und ihre Lösungen vorgestellt haben. Die Einführung einer der Lösungen benötigt in jedem Fall neue Investitionen. Der hier vorgestellte Wellen-Konfigurator ist natürlich die kostengünstigste Variante, mit einem riesigen Vorteil, dass es von Studierenden für Studierende gemacht wird. Die Studierenden der HAW haben es selbst in der Hand ein Werkzeug nach eigenen Wünschen und Bedürfnissen zu entwickeln. Das erfordert ein gewisses Maß an Interesse und Arbeit. Dafür hat man ein Werkzeug, das die eigene Arbeit erleichtert, viel Zeit spart und schließlich Spaß bringt. Es ist durch mehrmalige Tests bewiesen, dass beispielsweise, wenn man eine Welle in Form des einfachen Zylinders modelliert, spart man bis zur Hälfte der Zeit mit dem Wellen-Konfigurator gegenüber der konventionellen Methoden. Wenn man aber eine Zahnwelle modelliert, da beträgt die Ersparnis bis zu des Zwanzigfachen der Zeit, und das für einen geübten Menschen. Die Einarbeitung in das Thema wird in erster Linie durch die Aufteilung des Gesamtsystems und den großen Umfang der Creo API erschwert. Die Arbeit wird hoffentlich einige Unklarheiten beseitigen und die Weiterentwicklung vorantreiben.

Insgesamt kann man den Wellen-Konfigurator als Erfolg bezeichnen, er ist aber in dem aktuellen Zustand noch verbesserungswürdig. Er erfüllt zwar ganz seinen Zweck, wurde aber noch nicht anderen Nutzern vorgestellt. Daher ist es noch unklar welche Fehler oder Lücken in den Funktionalitäten oder der Bedienung entdeckt werden. Er soll also zuerst von einer Probandengruppe getestet werden. Sie muss feststellen ob der Wellen-Konfigurator tatsächlich so funktioniert, wie es gedacht wurde, und eventuell Verbesserungsvorschläge einbringen.

Es ist ungewiss, ob der Umfang von Funktionen und Elementen ausreichend ist. Daher wird es sich mit der Zeit zeigen, ob etwas ergänzt werden müsste.

Man kann jetzt mit Sicherheit sagen, dass vielleicht die grafische Benutzeroberfläche in mitgelieferten Creo GUI Editor, der zur Zeit aus unbekanntem Gründen nicht funktionsfähig ist, entwickelt werden könnte oder sollte. Es ist der Versuch wert zu testen, ob die neue Benutzeroberfläche dann in Creo Parametric integrierbar wäre.

Anhang dieses Beispiels können weitere Konfiguratoren, bspw. Getriebe oder Kupplungen, entwickelt werden. Für solche Konfiguratoren ist die Kollisionserkennungs-Funktion von großer Bedeutung.

Als nächstes Upgrade für den Wellen-Konfigurator wird die Ergänzung der Angaben durch die Auswahl von Werkstoffen, Toleranzen, Oberflächenbeschaffenheit usw. empfohlen.

Zuletzt zu erwähnen, die Zeichnungsableitung kann weiterverfolgt werden. Dafür soll man für bestimmte Wellen Schablonen entwickeln.

Abbildungsverzeichnis

Abbildung 2.1: Wellen-Generator: Fase, Gewinde [01]	10
Abbildung 2.2: Wellen-Generator: Keilnut, Nut für Sicherungsring [01]	10
Abbildung 2.3: Wellen-Generator: Punkte für Auflager und Belastungen [01]	11
Abbildung 2.4: Wellen-Generator: 3D-Vorschau [01]	11
Abbildung 2.5: Erzeugung einer Welle von eAssistant in Creo [02]	12
Abbildung 2.6: Erzeugung einer Passfeder in SolidGenius [03]	13
Abbildung 3.1: 3D-Modell: Hohlprofil	14
Abbildung 3.2: Skizze: Hohlprofil.....	15
Abbildung 3.3: 3D Modell: Zylinder.....	15
Abbildung 3.4: Skizze: Zylinder.....	16
Abbildung 3.5: 3D Modell: Kegel.....	16
Abbildung 3.6: Skizze: Kegel	17
Abbildung 3.7: 3D Modell: Keilwelle	17
Abbildung 3.8: Skizze: Keilwellenprofil.....	18
Abbildung 3.9: 3D Model: Zahnwelle mit Evolventenflanken.....	18
Abbildung 3.10: 3D Model: Polygonprofil P3G.....	19
Abbildung 3.11: Skizze: Polygonprofil P3G.....	19
Abbildung 3.12: 3D Model: Polygonprofil P4C.....	20
Abbildung 3.13: Skizze: Polygonprofil P4C	20
Abbildung 3.14: 3D Modell: Zylinder mit Zentrierbohrung.....	21
Abbildung 3.15: 3D Modell: Zylinder mit Radialbohrung.....	22
Abbildung 3.16: Skizze: Radialbohrung	22
Abbildung 3.17: 3D Modell: Zylinder mit Fase	23
Abbildung 3.18: Skizze: Fase	23
Abbildung 3.19: 3D Modell: Zylinder mit Rundung.....	24
Abbildung 3.20: Skizze: Rundung	24
Abbildung 3.21: 3D Modell: Zylinder mit Gewinde.....	25
Abbildung 3.22: Skizze: Gewinde	25
Abbildung 3.23: 3D Modell: Zylinder mit Gewindefreistich.....	26
Abbildung 3.24: Skizze: Gewindefreistich	26
Abbildung 3.25: 3D Modell: Zylinder mit Freistich.....	27
Abbildung 3.26: Skizze: Freistich Form E.....	27
Abbildung 3.27: Skizze: Freistich Form F.....	28
Abbildung 3.28: 3D Modell: Zylinder mit Passfedernut Form A.....	28
Abbildung 3.29: Skizze: Passfedernut Form A.....	29
Abbildung 3.30: 3D Modell: Zylinder mit Nut für Sicherungsringe	29
Abbildung 3.31: Skizze: Nut für Sicherungsringe.....	30
Abbildung 3.32: 3D Modell: Zylinder mit Nut für Sicherungsblech.....	30
Abbildung 3.33: Skizze: Nut für Sicherungsblech	31
Abbildung 4.1: Flussdiagramm: Komponente löschen.....	36
Abbildung 4.2: Flussdiagramm: Wellenabsatz löschen	37
Abbildung 4.3: Flussdiagramm: eine Komponente in Baugruppe einfügen.....	41
Abbildung 4.4: Flussdiagramm: UDF einfügen	45
Abbildung 4.5: Flussdiagramm: Gewindedarstellungen ändern	47

Quellenverzeichnis:

[01]:

<https://knowledge.autodesk.com/de/support/inventor-products/learn-explore/caas/CloudHelp/cloudhelp/2016/DEU/Inventor-Help/files/GUID-9BDDFABF-0A38-4A10-AFDC-4D5450A51DD6-htm.html>

[02]:

http://www.eassistant.eu/fileadmin/dokumente_web/eassistant/pdf/de/proe/Benutzung_ProE_d.pdf

[03]:

Bild aus einem YouTube Video von MIP: <https://www.youtube.com/watch?v=D8h0i4CaUgk>

[Ro/Ma 15]:

Roloff/Matek Maschinenelemente: Normung, Berechnung, Gestaltung
Wittel, Herbert, Muhs, Dieter, Jannasch, Dieter, Voßiek, Joachim
22., überarb. und erw. Aufl. Springer Vieweg 2015

[T. Z. 3 A.]:

Technisches Zeichnen
Susanna Labisch, Christian Weber
3.Auflage, Vieweg



Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Gemäß der Allgemeinen Prüfungs- und Studienordnung ist zusammen mit der Abschlussarbeit eine schriftliche Erklärung abzugeben, in der der Studierende bestätigt, dass die Abschlussarbeit „– bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit [(§ 18 Abs. 1 APSO-TI-BM bzw. § 21 Abs. 1 APSO-INGI)] – ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt wurden. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich zu machen.“

Quelle: § 16 Abs. 5 APSO-TI-BM bzw. § 15 Abs. 6 APSO-INGI

Dieses Blatt, mit der folgenden Erklärung, ist nach Fertigstellung der Abschlussarbeit durch den Studierenden auszufüllen und jeweils mit Originalunterschrift als letztes Blatt in das Prüfungsexemplar der Abschlussarbeit einzubinden.

Eine unrichtig abgegebene Erklärung kann -auch nachträglich- zur Ungültigkeit des Studienabschlusses führen.

Erklärung zur selbstständigen Bearbeitung der Arbeit

Hiermit versichere ich,

Name: _____

Vorname: _____

dass ich die vorliegende _____ – bzw. bei einer Gruppenarbeit die entsprechend gekennzeichneten Teile der Arbeit – mit dem Thema:

ohne fremde Hilfe selbstständig verfasst und nur die angegebenen Quellen und Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

- die folgende Aussage ist bei Gruppenarbeiten auszufüllen und entfällt bei Einzelarbeiten -

Die Kennzeichnung der von mir erstellten und verantworteten Teile der _____ ist erfolgt durch:

_____ Ort

_____ Datum

_____ Unterschrift im Original