

Die Visual Basic Programmierschnittstelle – eine experimentelle Untersuchung der Möglichkeiten

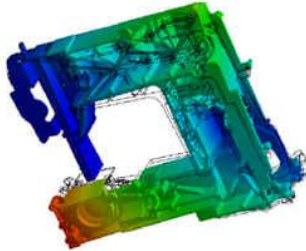
Dipl.-Ing. (FH) Christoph Kesselmanns
Dipl.-Ing. Marcin Humpa
Universität Duisburg-Essen
17. November 2010

Die Visual Basic Programmierschnittstelle – eine experimentelle Untersuchung der Möglichkeiten

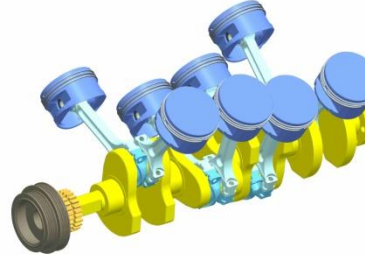
- > Einleitung
- > Motivation
- > Zugang über MS Office Programme (VBA)
- > Zugang über VB.Net Anwendungen
- > Zusammenfassung

Vorstellung Lehrstuhl RIKD

Strukturberechnung
und -optimierung



CAD



UNIVERSITÄT
D U I S B U R G
E S S E N

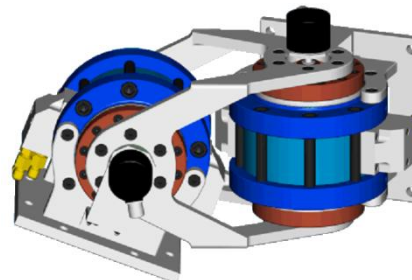
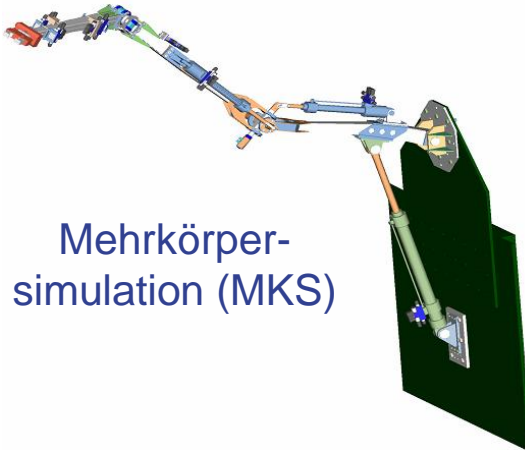
>>>>

ipe

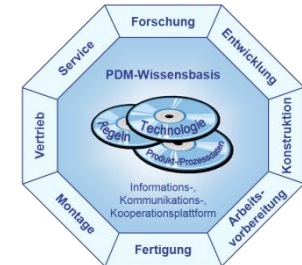
Institut für Produkt Engineering
Rechnereinsatz in der Konstruktion

<http://www.uni-due.de/cae>

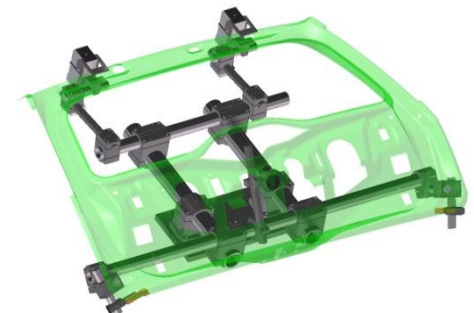
Mehrkörper-
simulation (MKS)



Integration von Gestaltung
und Berechnung

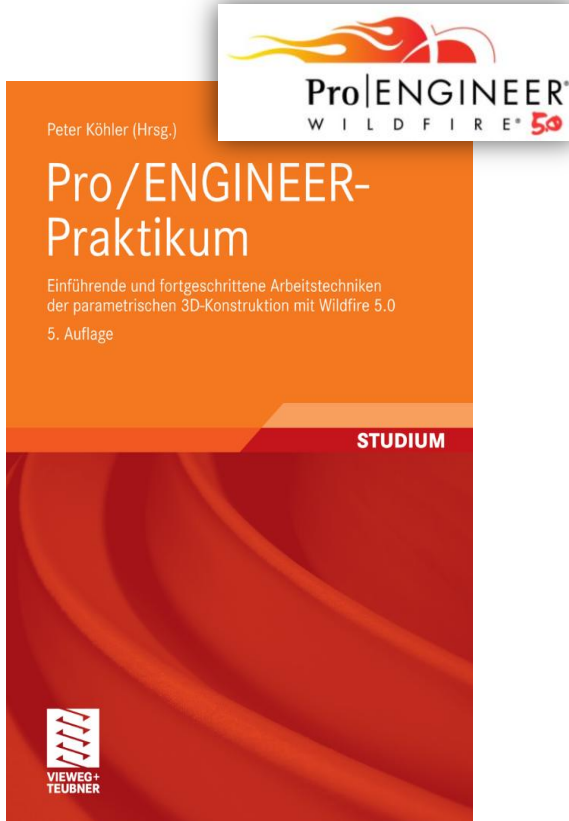


Produktdaten- und
Informationsmanagement
PDIM



Knowledge Based
Engineering (KBE)

Publikationen

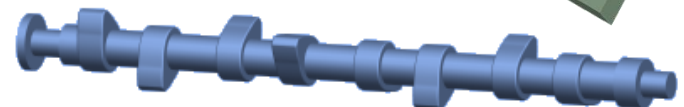
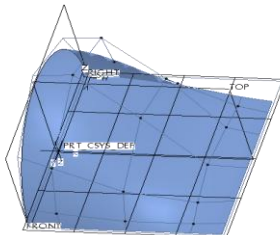
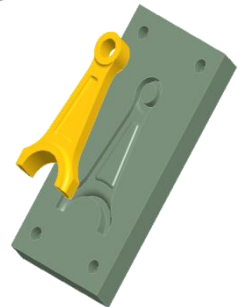
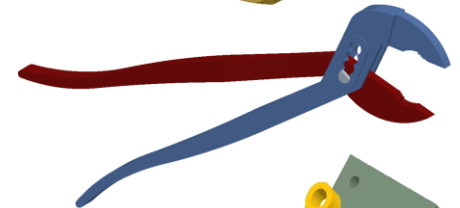
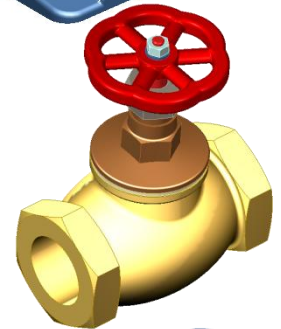
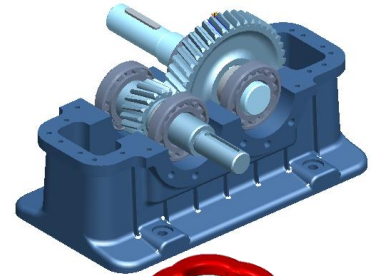


P. Köhler u.a.: Pro/ENGINEER-Praktikum Vieweg-Verlag

1. Auflage 1999,
2. aktualisierte Auflage 2000,
3. vollständig überarbeitete Auflage 2003
4. erweiterte Auflage 2006
5. Auflage 2010

Inhalt

- Einführung in die Arbeit mit Wildfire 5.0
- Modellreferenzen
- Skizzieren
- Bauteilmodellierung
- Baugruppenmodellierung
- Zeichnungsableitung aus dem 3D-Modell
- Arbeitstechniken zur Produktoptimierung



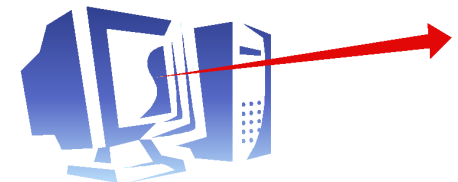
Kooperationen

- ✓ Analyse und Optimierung betrieblicher Konstruktions- und Entwicklungsprozesse
- ✓ Integration unternehmensspezifischen Wissens in den rechnergestützten Entwicklungs- und Konstruktionsprozess im Rahmen des Knowledge Based Engineering (KBE)
- ✓ Produktbezogene Integration von Gestaltung und Berechnung
- ✓ Mitwirkung beim Aufbau firmenspezifischer Informations- und Produktdatenmanagementsysteme
- ✓ Entwicklung von Methoden zur Qualitätssicherung digitaler Produkt- und Prozessmodelle
- ✓ Berechnungen im Bereich der Strukturanalyse (statisch/dynamisch), Mehrkörpersimulation, etc.
- ✓ Schulungen und Seminare zu CAx-Technologien



Prof. Dr.-Ing. Dipl.-Math. Peter Köhler
Universität Duisburg-Essen
Fakultät für Ingenieurwissenschaften
Institut für Produkt Engineering

47057 Duisburg
Tel.: +49 (0)203 379-2790
Fax: +49 (0)203 379-4379
e-mail: peter.koehler@uni-due.de
<http://www.uni-due.de/cae/>



Die Visual Basic Programmierschnittstelle – eine experimentelle Untersuchung der Möglichkeiten

- > Einleitung
- > **Motivation**
- > Zugang über MS Office Programme (VBA)
- > Zugang über VB.Net Anwendungen
- > Zusammenfassung

Allgemeines zur API (Application Programming Interface)

> Potentiale der API

- Automatisierung von Konstruktionsprozessen
z.B. automatisches Zusammenstellen von Baugruppen oder die Steuerung von Familientabellen
- Wissensintegration z.B. Gestaltung von Schleifen, Fallunterscheidungen oder Regeln
- Anbindung externer Applikationen und Software
- Automatische Anpassung von Pro/ENGINEER Modellen
- Reaktion auf Ereignisse während des Konstruktionsprozesses
- Prüfroutinen um Konstruktionsstandards zu überwachen
- ...

> Wie funktioniert die API?

- Die Applikation (COM-Client) instanziiert eine COM-Komponente in einem COM-Server (Pro/ENGINEER API) und nutzt die Funktionalität des Objektes über COM-Interfaces.
- Programmanbindung auf Quellcodeebene (direkter Zugriff auf Funktionen, Methoden und Eigenschaften der Objekte)

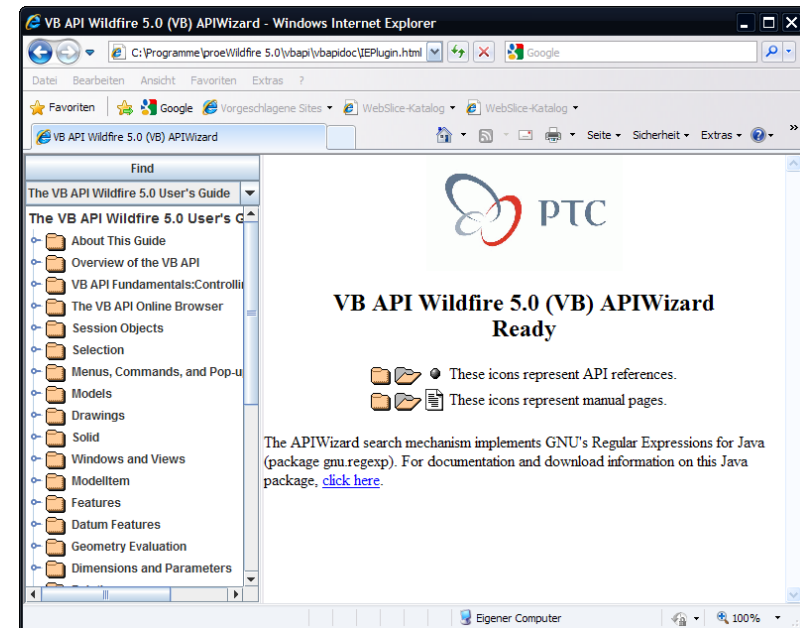
Allgemeines zur API (Application Programming Interface)

> Grenzen der VB API

- Es kann immer nur eine Verbindung zwischen einer Applikation und einer Pro/ENGINEER Sitzung bestehen

> Einarbeitung

- Grundkenntnisse der Pro/ENGINEER Modellierung und der objektorientierten Programmierung nötig
- Mitgelieferter „User's Guide“ und der „APIWizard“ ermöglichen eine rasche Einarbeitung und stellen wichtige Suchfunktionen zur Verfügung
- Weiterhin stehen Beispielprojekte zur Verfügung
- Zur allgemeinen VB-Programmierung -> Internet



> Applikationshierarchie

- Die Regeln der OOP fordern die Einhaltung einer vorgegebenen Hierarchie um Objekte ansprechen zu können. Die Applikation muss bis zur Tiefe des anzusprechenden Objektes vordringen.

Applikationshierarchie

> Achsen eines Bohrungsfeatures aus aktivem Modell filtern

1. Verbindung zu Pro/ENGINEER erzeugen

```
Dim cAC As CCpfcAsyncConnection  
Dim asyncConnection As IpfcAsyncConnection = Nothing  
  
cAC = New CCpfcAsyncConnection()  
asyncConnection = cAC.Connect(Nothing, Nothing, Nothing, Nothing)
```

2. Ansprechen der aktuellen Sitzung

```
Dim session As IpfcBaseSession  
session = asyncConnection.Session
```

3. Momentan aktives Modell ansprechen

```
Dim model As IpfcModel  
model = session.CurrentModel()
```

4. Feature des Modells auflisten

```
Dim modelitems As IpfcModelItems  
Dim Feature As IpfcFeature  
Dim Axes As IpfcModelItems
```

```
modelitems = model.listitems(EpfcModelItemType.EpfcITEM_FEATURE)
```

5. Filtern der Bohrungsfeature

```
For i = 0 To modelitems.Count - 1  
    Feature = modelitems(i)
```

6. Achse der Bohrung in der Variable „Axes“ gespeichert

```
    If Feature.FeatType = EpfcFeatureType.EpfcFEATTYPE_HOLE Then  
        Axes = Feature.ListSubItems(EpfcModelItemType.EpfcITEM_AXIS)  
    End If
```

```
Next
```

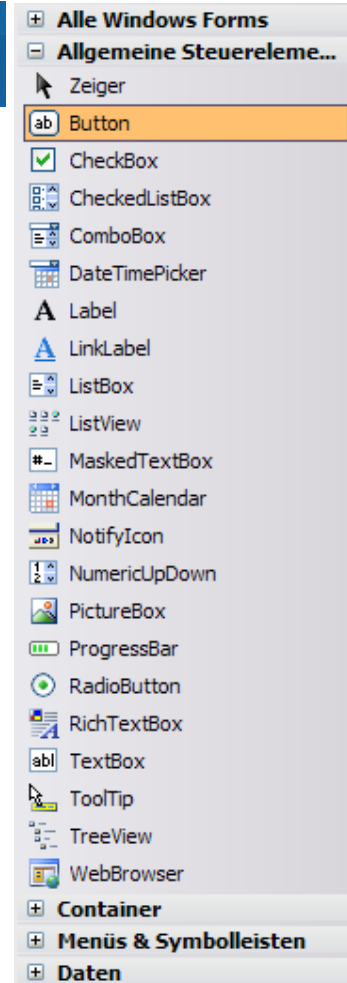
Entwicklungsumgebungen

> VB.Net 2008 Applikationen

- „Windows Forms“ Anwendung mit typischen Elementen für Ein- und Ausgabe von Daten

+ Automatisches Vollständigen
Syntaxprüfung
Form und Dialogerstellung
Laufzeitfehlerbehandlung

- Anfangs unübersichtlicher da
größerer Funktionsumfang

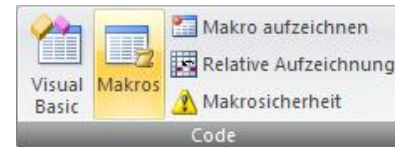


> VBA Makros - Visual Basic for Applications

- Zugang über MS Office Programme wie Excel, Word, Access, Visio etc.

+ Da eingeschränkter, leichter
erlernbar und übersichtlicher.
Ist in MS Office integriert, somit
keine zusätzliche Software nötig.

- Veraltet, trotzdem in Office
2010 integriert.
Keine Zeilennummerierung
Siehe Vorteile DotNet



Die Visual Basic Programmierschnittstelle – eine experimentelle Untersuchung der Möglichkeiten

- > Einleitung
- > Motivation
- > Zugang über MS Office Programme (VBA)
- > Zugang über VB.Net Anwendungen
- > Zusammenfassung

Beispiel: MS Word - Bericht zu Pro/ENGINEER-Teilen erzeugen

- > Für viele Teile, alle Parameter (Name, Typ, Wert) die Masse und den Oberflächeninhalt auslesen und auf einem Datenblatt zusammenfassen

The screenshot displays two overlapping windows. The background window is Microsoft Word, showing a document titled 'Bericht'. It contains a list of parameters to be extracted from Pro/ENGINEER parts, such as Name, Speicherort, Typ, Volumen, Dichte, Masse, and Oberfläche. Below this list is a table with columns 'Parametername' and 'Typ'. The foreground window is the Microsoft Visual Basic Editor, showing a VBA macro named 'Makro1'. The macro code is as follows:

```
Sub Makro1()  
    ' Makro1 Makro  
End Sub
```

A 'Verweise - Normal' dialog box is open, showing a list of available references. The 'Pro/E VB API Types Library for Pro/E Wildfire 5.0' is selected. The dialog also shows the path 'C:\Programme\proeWildfire 5.0\obj\lpfidscom.exe' and the language 'Voreinstellung'. At the bottom of the Visual Basic Editor, there are three panes: 'Direktbereich', 'Lokal', and 'Überwachungsausdrücke'.

> Vorgehensweise:

- Word Makro erstellen
- Verweis auf Pro/ENGINEER API festlegen
- ...

Beispiel: MS Word - Bericht zu Pro/ENGINEER-Teilen erzeugen

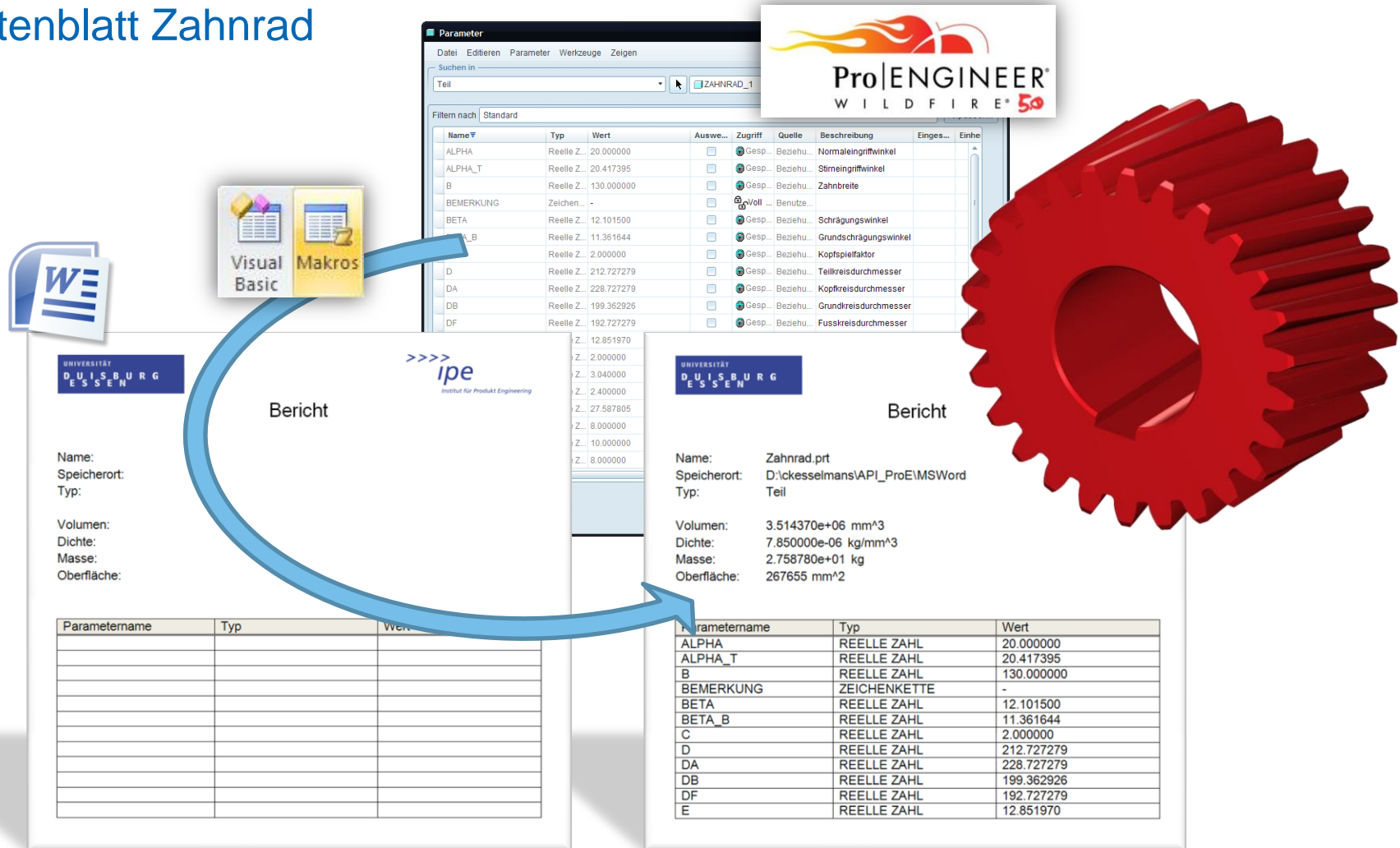
> Vorgehensweise:

- Verbindung zu Pro/ENGINEER Sitzung
 - Funktion: `GetModelFromFileName(„Name“)`
- Teil aus Verzeichnis ansprechen
 - Funktion: `item(index)`
- Parameter identifizieren (Schleife)
 - Funktion: `item(index)`
- Parameterwert auslesen
 - Funktion: `GetScaledValue`
- Wert als String an Textdokument übergeben
- Masse auslesen
 - Funktion: `solid.GetMassProperty(Nothing)`
 - Eigenschaft: `solidproperties.Mass`
- Oberflächeninhalt äquivalent zu Masse

```
Sub Makro1()  
.  
    Dim asyncConnection As IpfcAsyncConnection  
    Dim cAC As New CCpfcAsyncConnection  
    Dim session As IpfcBaseSession  
    Dim model As IpfcModel  
    Dim ParaOwner As IpfcParameterOwner  
    Dim parameter As IpfcParameter  
    Dim parameters As IpfcParameters  
    Dim Value As IpfcParamValue  
    Dim solid As IpfcSolid  
    Dim solidproperties As IpfcMassProperty  
  
    Set asyncConnection = cAC.Connect(Null, Null, Null, Null)  
    Set session = asyncConnection.session  
    Set model = session.GetModelFromFileName("Teil.prt")  
    Set ParaOwner = model  
    Set parameters = ParaOwner.ListParams  
  
    For i = 0 To parameters.Count - 1  
        Set parameter = parameters.Item(i)  
        Set Value = parameter.GetScaledValue  
        Selection.TypeText Text:=CStr(Value.DoubleValue)  
        Selection.TypeParagraph  
    Next  
  
    Set solid = model  
    Set solidproperties = solid.GetMassProperty(Null)  
    Selection.TypeText Text:=CStr(solidproperties.Mass)  
    Selection.TypeParagraph  
  
End Sub
```

Beispiel: MS Word - Bericht zu Pro/ENGINEER-Teilen erzeugen

> Datenblatt Zahnrad



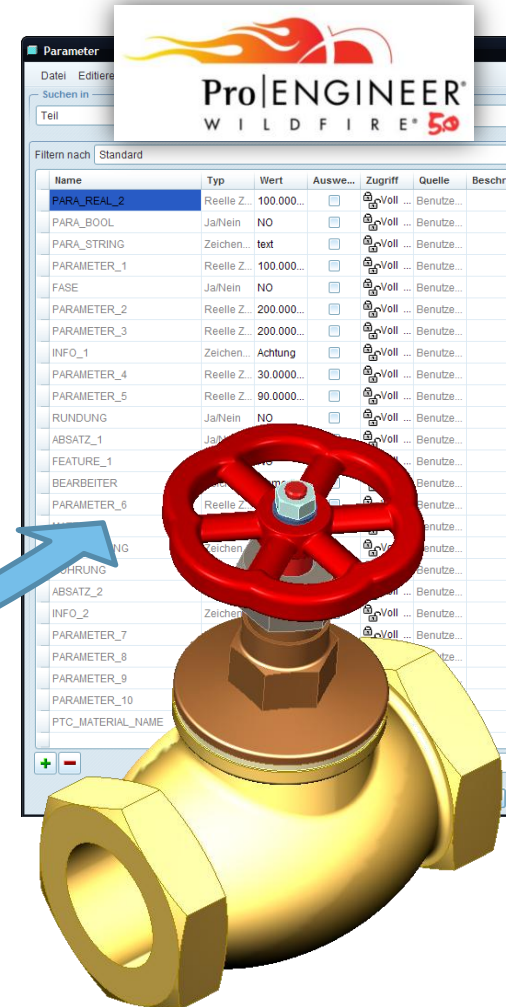
Beispiel: MS Excel - Parameter im Teil aus Tabelle erzeugen (nicht editieren)

> Parameter (Name, Typ, Wert) liegen in einer Excel Tabelle vor und sollen als Teileparameter erzeugt werden

The screenshot shows the Microsoft Excel interface with the 'Entwicklertools' (Developer) tab active. The 'Code' window displays a VBA macro named 'makro1' that reads data from an Excel table and creates parameters in a CAD system. The table data is as follows:

Name	Typ	Wert
Parameter_1	Reelle Zahl	100
Fase	Boolean	false
Parameter_2	Reelle Zahl	200
Parameter_3	Reelle Zahl	200
Info_1	String	Achtung
Parameter_4	Reelle Zahl	30
Parameter_5	Reelle Zahl	90
Rundung	Boolean	false
Absatz_1	Boolean	true
Feature_1	Boolean	false
Bearbeiter	String	Name
Parameter_6	Reelle Zahl	70
Material	String	Stahl
Verwendung	String	sonderteil
Bohrung	Boolean	false
Absatz_2	Boolean	true
Info_2	String	text
Parameter_7	Reelle Zahl	150
Parameter_8	Reelle Zahl	180
Parameter_9	Reelle Zahl	20
Parameter_10	Reelle Zahl	9

```
Sub makro1()  
    Dim asyncConnection As IpfcAsyncConnection  
    Dim cAC As New CCpfcAsyncConnection  
    Dim session As IpfcBaseSession  
    Dim model As IpfcModel  
    Dim ParaOwner As IpfcParameterOwner  
    Dim Value As IpfcParamValue  
    Dim CParamvalue As CMpfcModelItem  
    Dim Wert As String  
    Dim Name As String  
    Dim Typ As String  
  
    Set asyncConnection = cAC.Connect(Null, Null, Null, Null)  
    Set session = asyncConnection.session  
    Set model = session.GetModelFromFileName("Teil.prt")  
    Set ParaOwner = model  
    Set CParamvalue = New CMpfcModelItem  
  
    For col = 2 To 4  
  
        Name = Worksheets(1).Cells(col, 1).Value  
        Typ = Worksheets(1).Cells(col, 2).Value  
        Wert = Worksheets(1).Cells(col, 3).Value  
  
        Select Case Typ  
  
            Case "Reelle Zahl"  
                Set Value = CParamvalue.CreateDoubleParamValue(Wert)  
  
            Case "Boolean"  
                Set Value = CParamvalue.CreateStringParamValue(Wert)  
  
            Case "Boolean"  
                Set Value = CParamvalue.CreateBoolParamValue(Wert)  
  
        End Select  
  
        Set para = ParaOwner.CreateParam(Name, Value)  
  
    Next col  
End Sub
```



Die Visual Basic Programmierschnittstelle – eine experimentelle Untersuchung der Möglichkeiten

- > Einleitung
- > Motivation
- > Zugang über MS Office Programme (VBA)
- > Zugang über VB.Net Anwendungen
- > Zusammenfassung

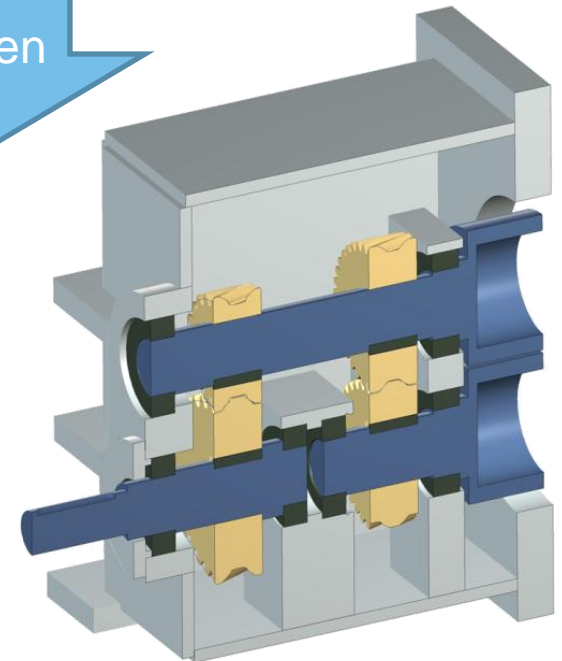
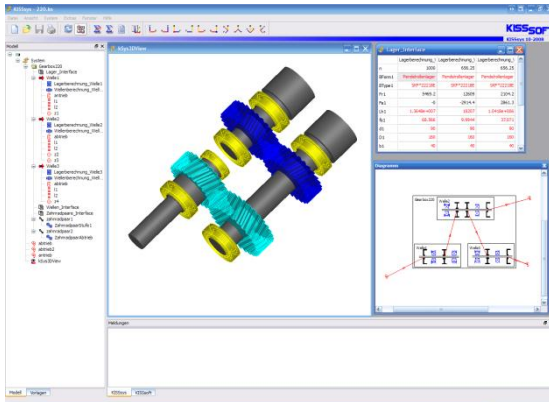
Projekt: Automatisierte Getriebeableitung mit Anbindung an Berechnungssoftware

> Anforderungen

- Getriebetypen: mehrstufige (1-3) Stirnrad- oder Kegelradgetriebe mit Leistungen bis 300kW
- Benötigte Funktionalität der Applikation:
 - Eingangsdaten erfassen
 - mögliche Übersetzungsparameter vorschlagen
 - XML-Dateien der Getriebeberechnungssoftware erzeugen und auslesen
 - automatisiert mit gewonnener Datenbasis CAD Modelle aufbauen -> API

Anforderungen
API

- Parameter editieren
- Familientabellen editieren
- Baugruppe zusammenbauen
- Kollisionsprüfung durchführen



Ablauf einer Getriebeauslegung

1. Vorgabe der Eingangsdaten



Getriebe Konfigurator

1. Allgemeine Eingaben 2. Mäding-Berechnungen 3. ProE Modell 4. Datenspeicherung

Eingabe

Antriebsdrehzahl: 1400 min⁻¹

Abtriebsdrehzahl: 980 min⁻¹

Leistung: 160 kW

Soll Übersetzung berechnen

zul. Schrägungswinkel von 10 bis 20 [Grad]

☒ Ganzzahlige Einzelübers. unterdrücken

Neue Getriebeliste mit Excel erzeugen

Getriebeliste übernehmen

Grenzdurchmesser-Teilkreis für Ritzelwelle: 200 [mm]

Kunden- Auftragsnummer: 4711-0815

Arbeitsverzeichnis: D:\ckesselmans\SMS_Projekt\Gro

Ausgabe

Soll - Übersetzung: 1,428

mögl. Übers. Modul 6: 0,411

mögl. Übers. Modul 8:

Abweichung: 71,18

z1: 51

z2: 21

beta: 16,260204708312

Modul: 6 [mm]

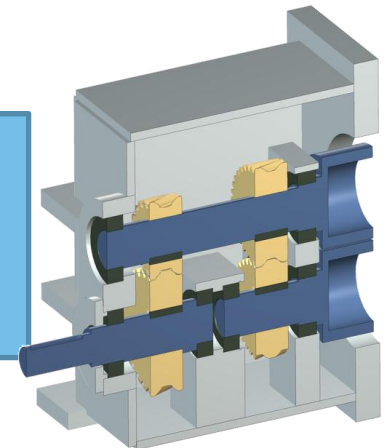
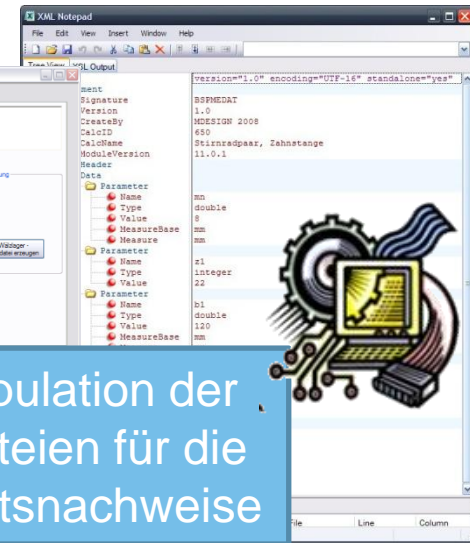
Ritzelwelle oder Welle Zahnrad Kombination: Welle-Ritzel Kombination

Program verlassen

2. Grobauslegung Zahnradparametern

4. Wenn eine vollständige Datenbasis vorliegt, wird automatisch eine neue Getriebevariante erzeugt

3. Manipulation der XML-Dateien für die Festigkeitsnachweise



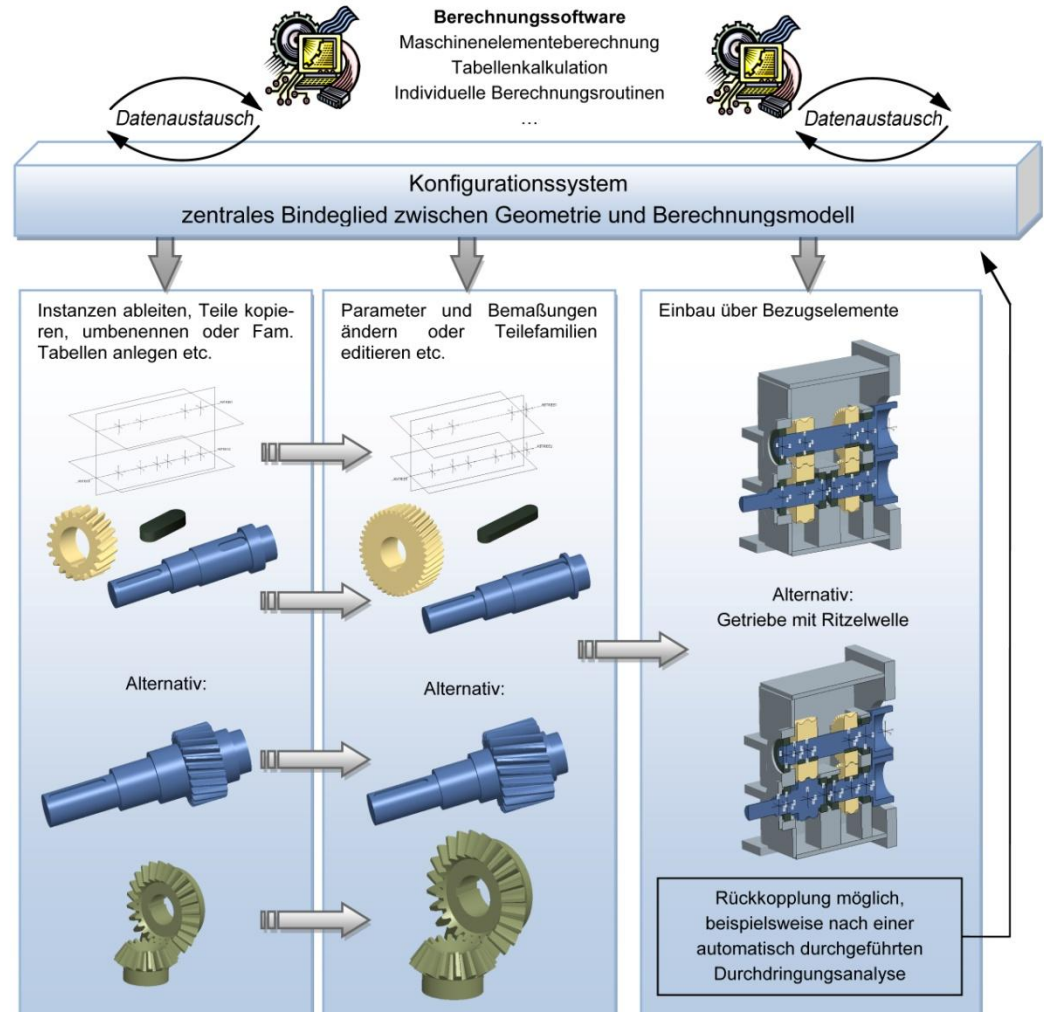
Benötigte API-Funktionalitäten zum Aufbau eines Getriebes

> Einzelteile editieren

- Teile identifizieren und ansprechen
- Parameter ändern
- Teile kopieren
- Familientabellen anlegen, ergänzen
- UDF's applizieren

> Baugruppen zusammenstellen

- Auswahl der Teile- bzw. Teilefamilien
- Strukturdaten (z.B. KOS) editieren
- Teile- und Baugruppenreferenzen (Skelett) identifizieren
- Einbaureferenzen festlegen
- Regenerierung der Baugruppe
- Denkbar: Durchdringungsprüfung



Beispiel: Feature aus vordefinierten UDF's erzeugen

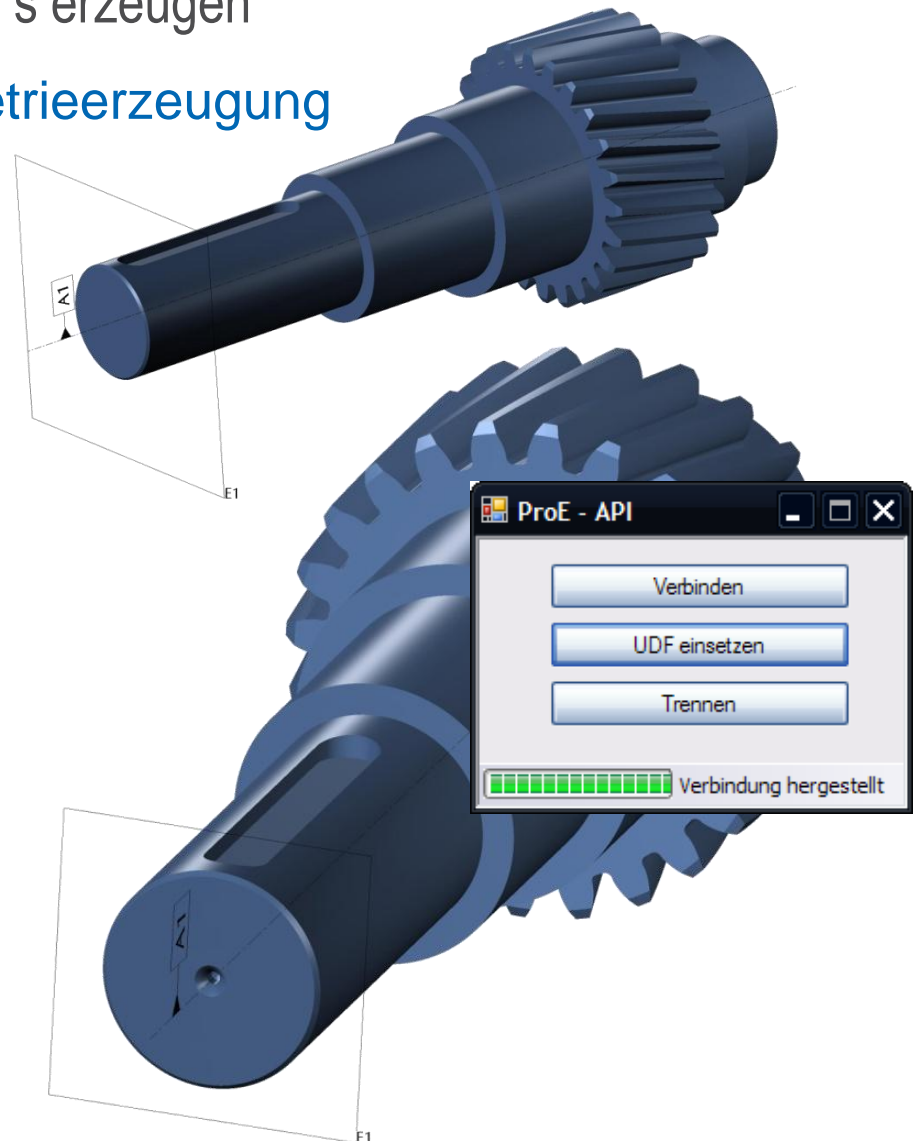
> Zugang zur automatischen Geometrieerzeugung

> Drei Möglichkeiten zur Referenzauswahl:

- Interaktiv mit Bedienereingaben
- Automatisch über Programmcode
- Semi-Interaktiv (Kombination)

> Beispiel Zentrierbohrung:

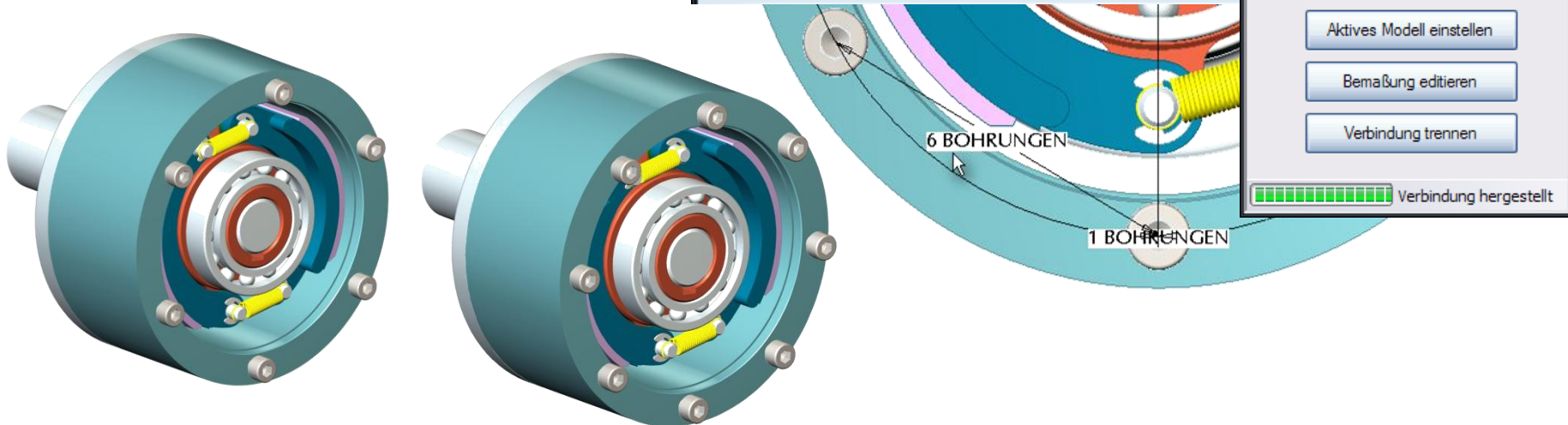
- UDF manuell erstellt
- Referenzen (E1 und A1) sind im Zielteil bekannt



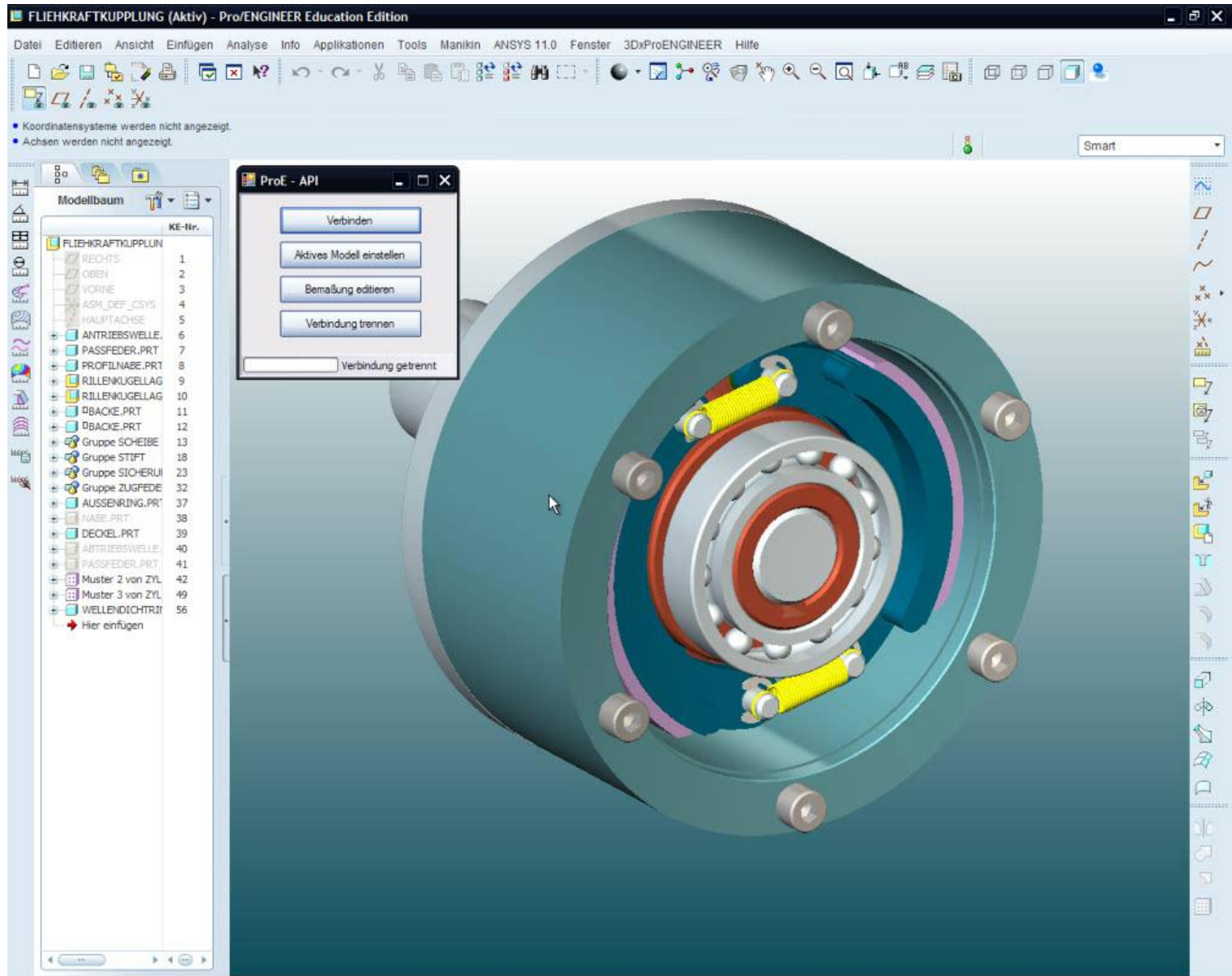
Beispiel: Benutzerselektion - Bemaßungen ändern

> Während der Laufzeit wird der Benutzer aufgefordert ein Feature zu selektieren, dann werden Bemaßungen angezeigt und mit neuen Werten aus dem Programm aktualisiert.

- Aktuelle Baugruppe ansprechen
- Featureselektion initiieren
- Bemaßungsauswahl einleiten
- Neuen Wert übergeben
- Regenerieren



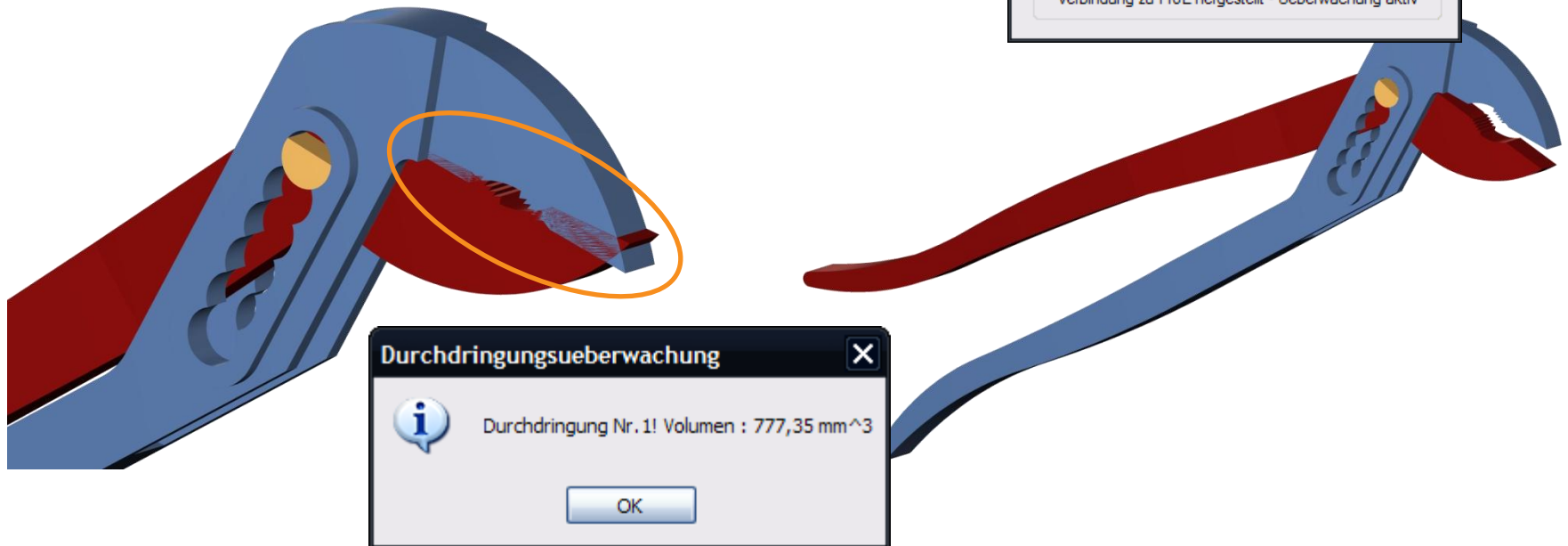
Zugang über VB.Net Anwendungen



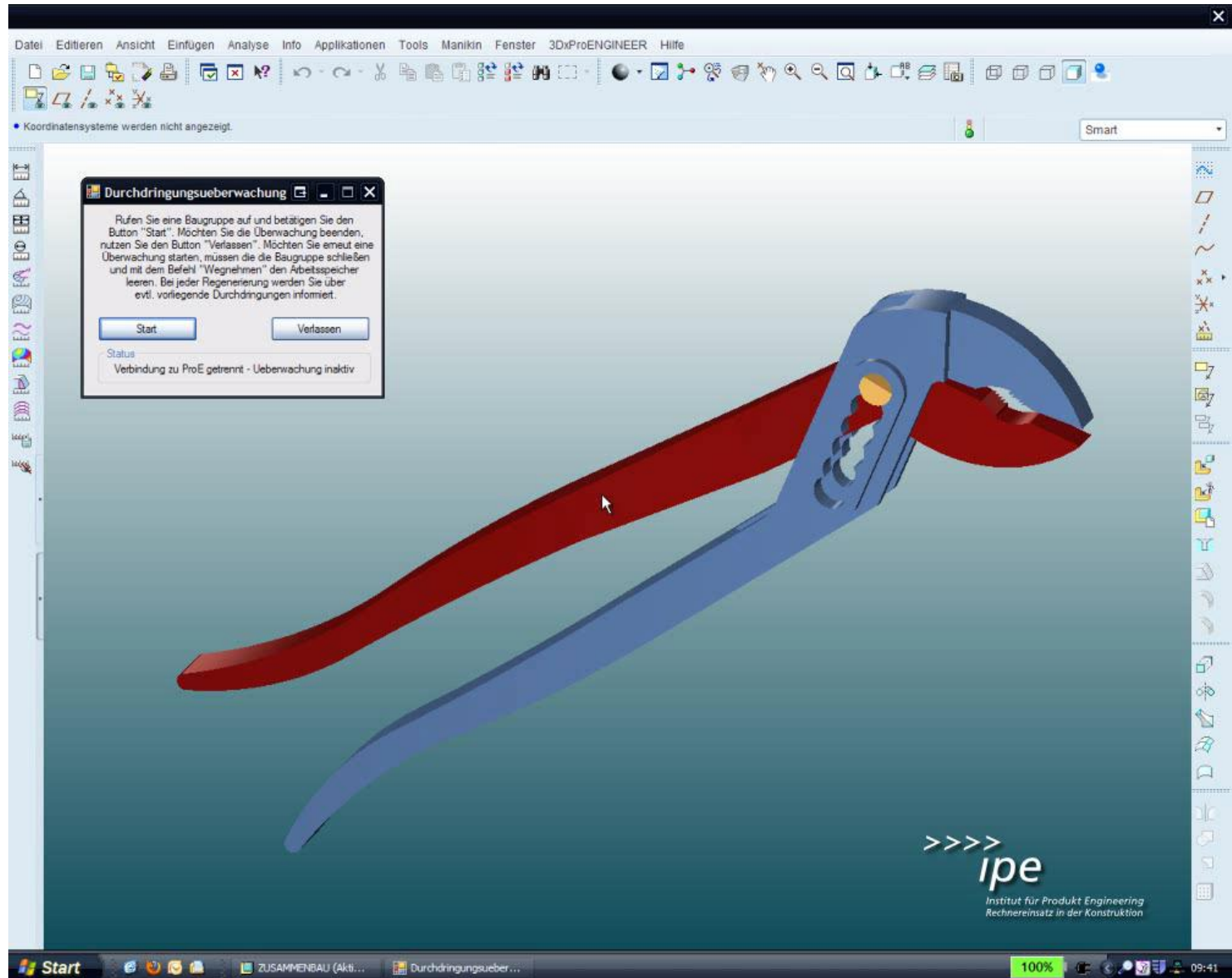
Beispiel: Auf Ereignisse reagieren - Durchdringungsüberwachung

> Beispiel: Automatische Durchdringungsanalyse nach dem Regenerieren

- Aktive Baugruppe ansprechen
- Sogenannten „Action Listener“ applizieren
- Nach einer Regenerierung springt die Applikation in eine Routine, die eine Durchdringungsanalyse auslöst
- Anzeigen der sich durchdringenden Körper und des Volumens



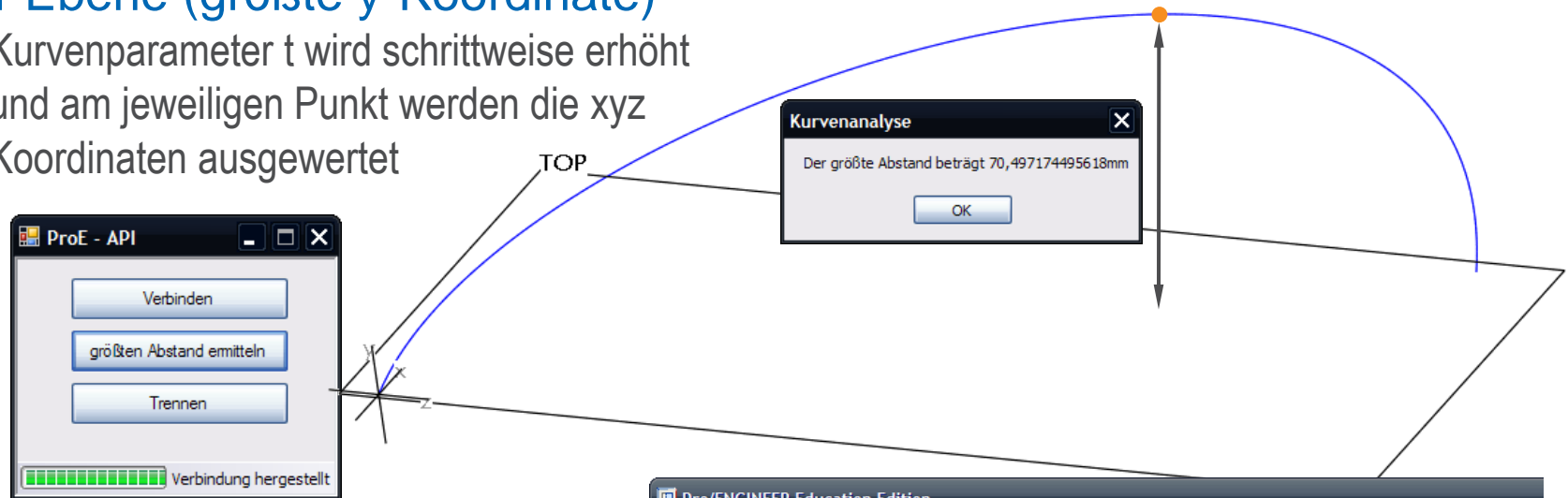
Zugang über VB.Net Anwendungen



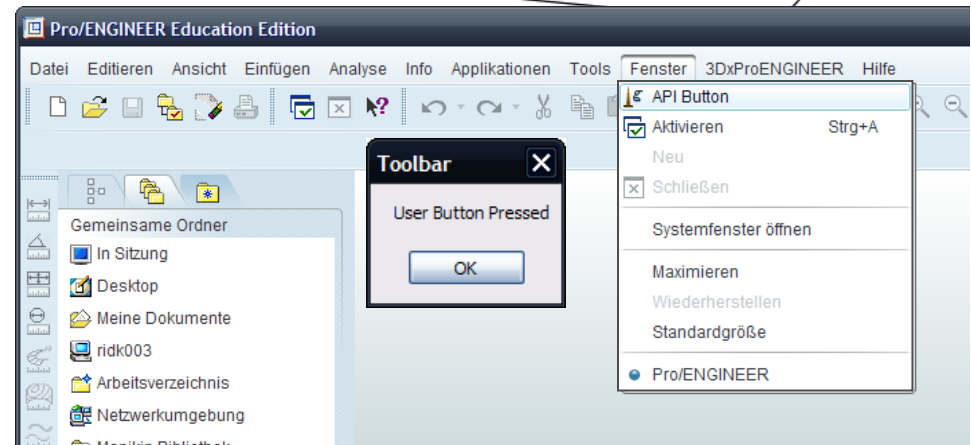
Beispiel: Kurvenanalyse und Menüeintrag

> Automatische Ermittlung des größten Abstands zwischen einer Kurve und einer Ebene (größte y-Koordinate)

- Kurvenparameter t wird schrittweise erhöht und am jeweiligen Punkt werden die xyz Koordinaten ausgewertet

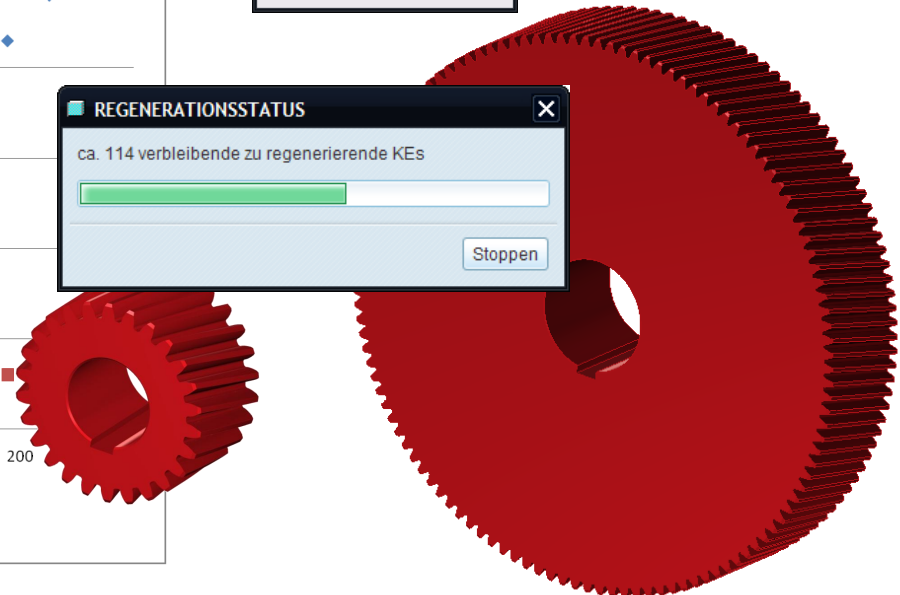
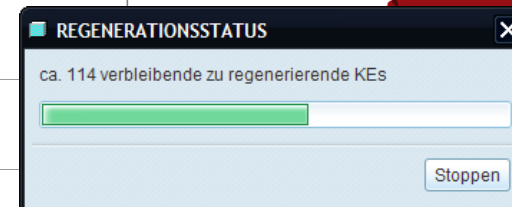
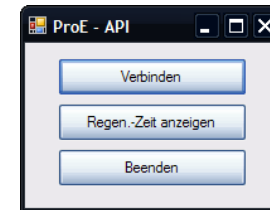
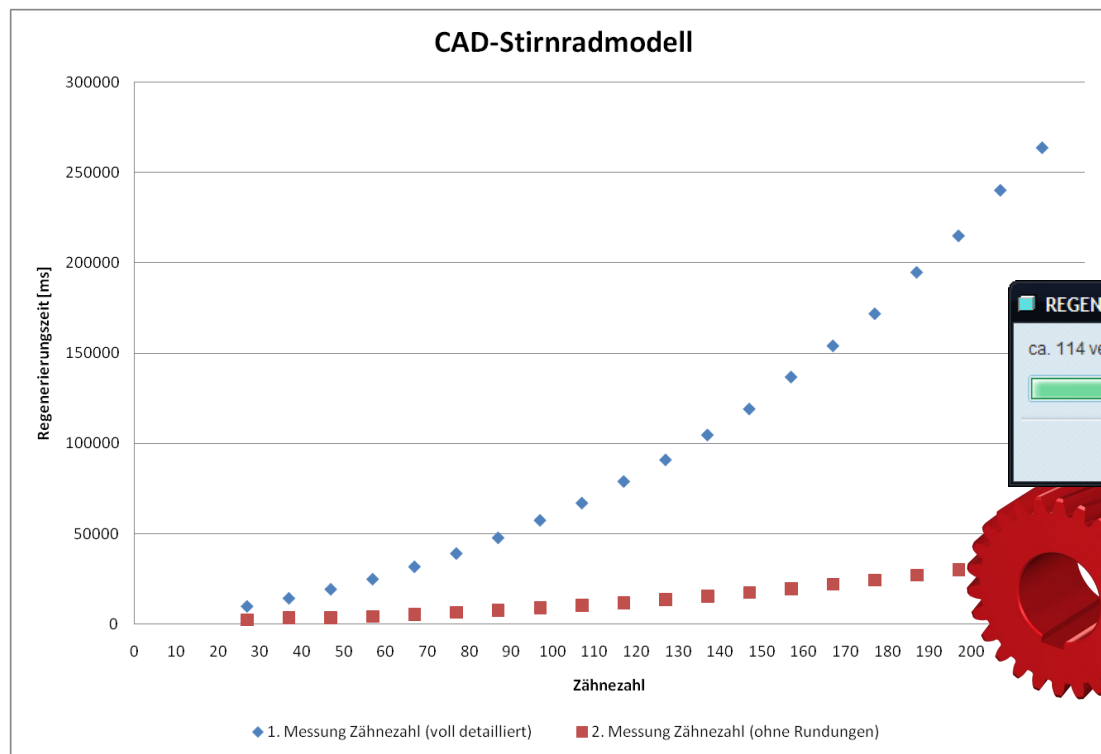


> Menüeinträge zur Laufzeit des Programms generieren und deren Auswahl überwachen



Beispiel: Zeitmessung

- > Welcher Zusammenhang besteht zwischen der Musteranzahl und der Regenerierungszeit in einem Teiledokument?
 - Ermittlung der Differenz der Systemzeit vor und nach der Regenerierung



Die Visual Basic Programmierschnittstelle – eine experimentelle Untersuchung der Möglichkeiten

- > Einleitung
- > Motivation
- > Zugang über MS Office Programme (VBA)
- > Zugang über VB.Net Anwendungen
- > Zusammenfassung

- > Auch wenn Pro/ENGINEER bereits viele Möglichkeiten und Funktionen zum intelligenten Modellaufbau bietet, kommt es vor, dass über eine benutzerspezifische Anpassung des Systems weitere signifikante Vorteile gewonnen werden können.
- > Kombination der vorgestellten Möglichkeiten kann zur Lösung anderer Problemstellungen beitragen.
- > Es stehen noch weitere Funktionen zur Verfügung z.B.:
 - Export – Import von Austauschformaten
 - Windchill Anbindungen
 - Vereinfachte Darstellungen
 - Beziehungen
 - Bezugselemente
 -



Vielen Dank für Ihr Interesse!