



# **Die Visual Basic Programmierschnittstelle – eine experimentelle Untersuchung der Möglichkeiten**

A decorative graphic in the bottom right corner consists of a dense, overlapping arrangement of stylized leaf or petal shapes. The shapes are rendered in thin outlines and come in various sizes and colors, including blue, green, and yellow, creating a organic, flowing pattern.

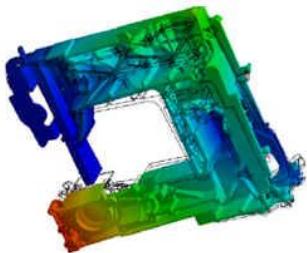
Dipl.-Ing. (FH) Christoph Kesselmans  
Dipl.-Ing. Marcin Humpa  
Universität Duisburg-Essen  
17. November 2010

## Die Visual Basic Programmierschnittstelle – eine experimentelle Untersuchung der Möglichkeiten

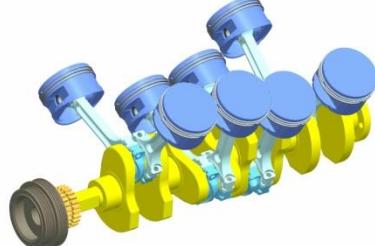
- > Einleitung
- > Motivation
- > Zugang über MS Office Programme (VBA)
- > Zugang über VB.Net Anwendungen
- > Zusammenfassung

## Vorstellung Lehrstuhl RIKD

Strukturberechnung  
und -optimierung



CAD

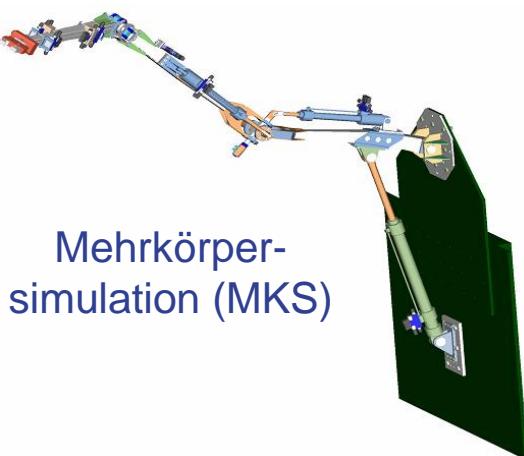


UNIVERSITÄT  
DUISBURG  
ESSEN

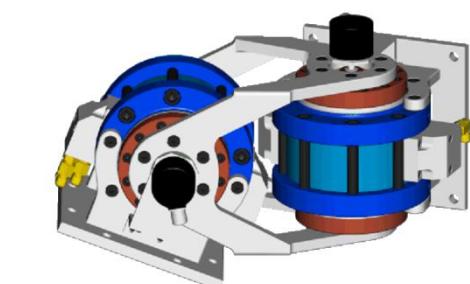
>>>  
ipe

Institut für Produkt Engineering  
Rechnereinsatz in der Konstruktion

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



Mehrkörper-  
simulation (MKS)



Integration von Gestaltung  
und Berechnung

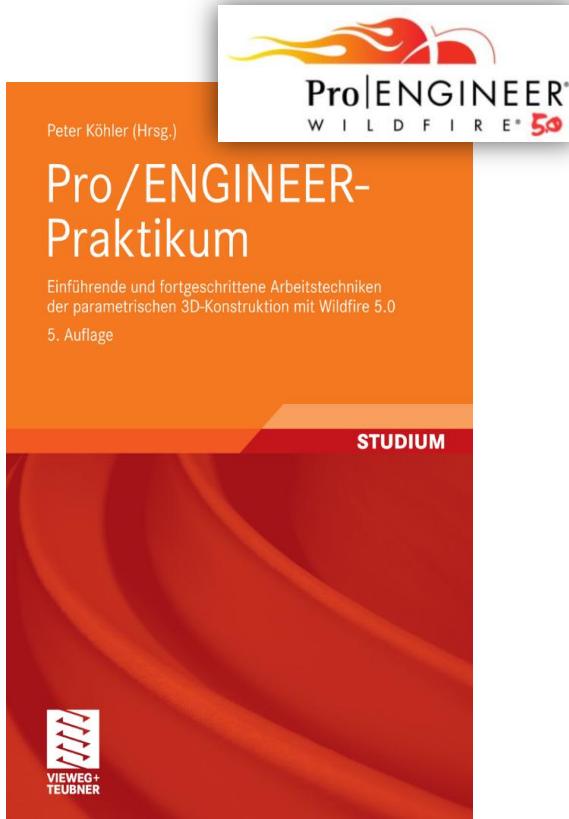


Produktdaten- und  
Informationsmanagement  
PDIM



Knowledge Based  
Engineering (KBE)

## Publikationen

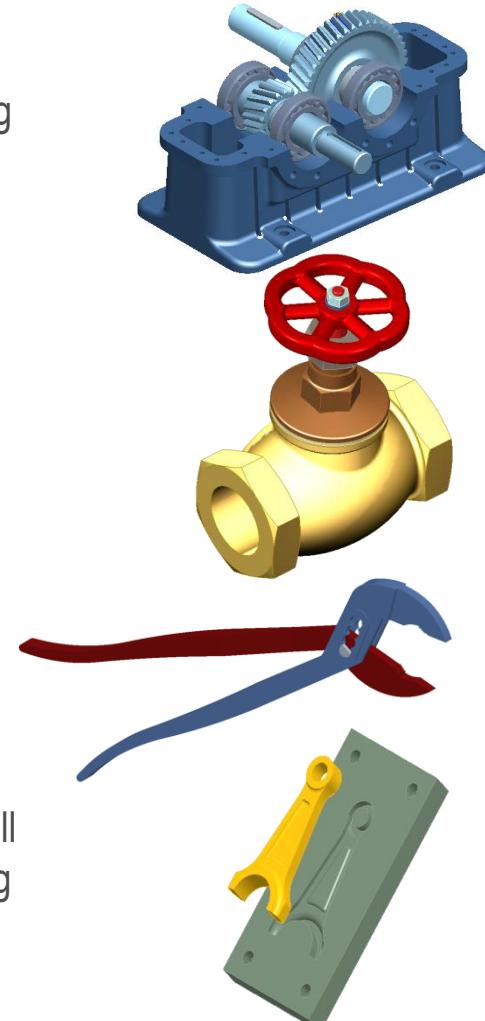
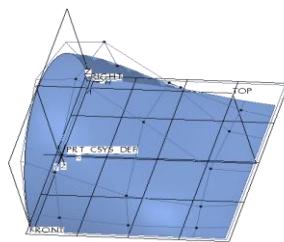


P. Köhler u.a.:  
**Pro/ENGINEER-Praktikum** Vieweg-Verlag

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](mailto: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) instanziert 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)

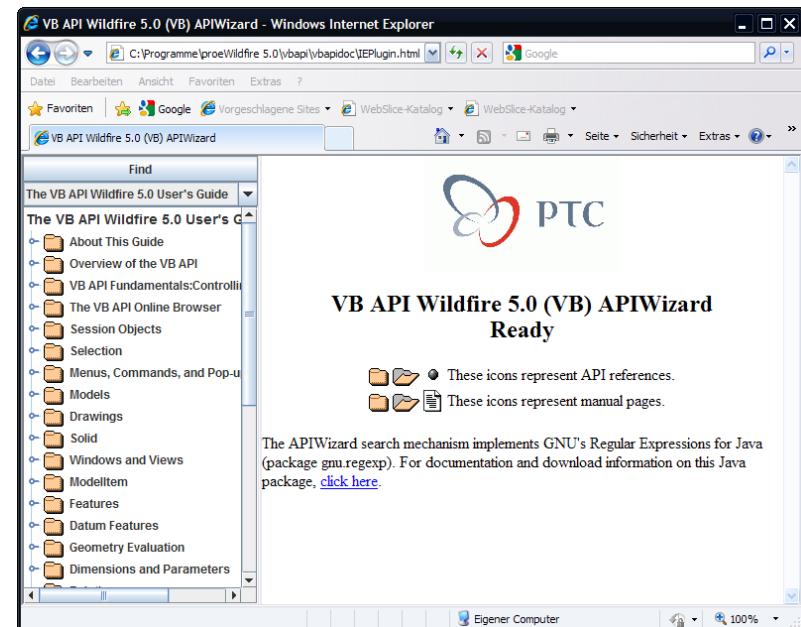
## Allgemeines zur API (Application Programming Interface)

### > 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.EpfcFEATTTYPE_HOLE Then
    Axes = Feature.ListSubItems(EpfcModelItemType.EpfcITEM_AXIS)
End If
```

Next

## Entwicklungsumgebungen

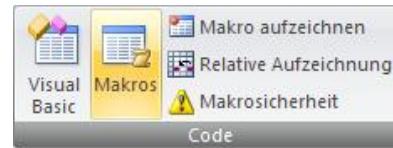
### > VB.Net 2008 Applikationen

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

+ Automatisches Vollständigen Syntaxprüfung  
Form und Dialogerstellung  
Laufzeitfehlerbehandlung



- Anfangs unübersichtlicher da größerer Funktionsumfang

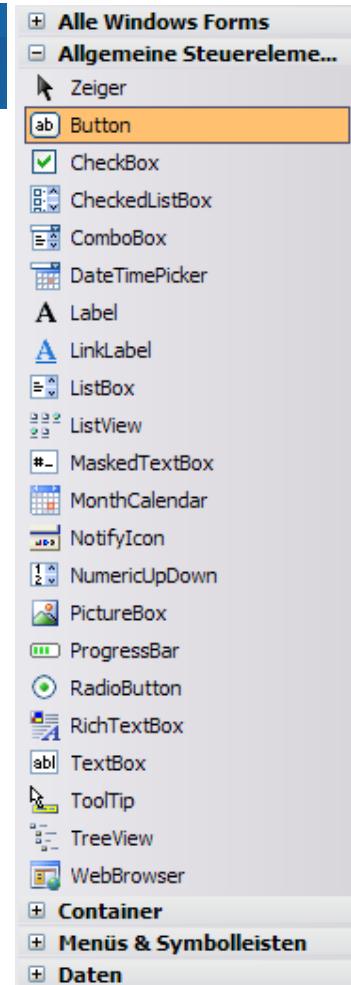


### > VBA Makros - Visual Basic for Applications

- 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

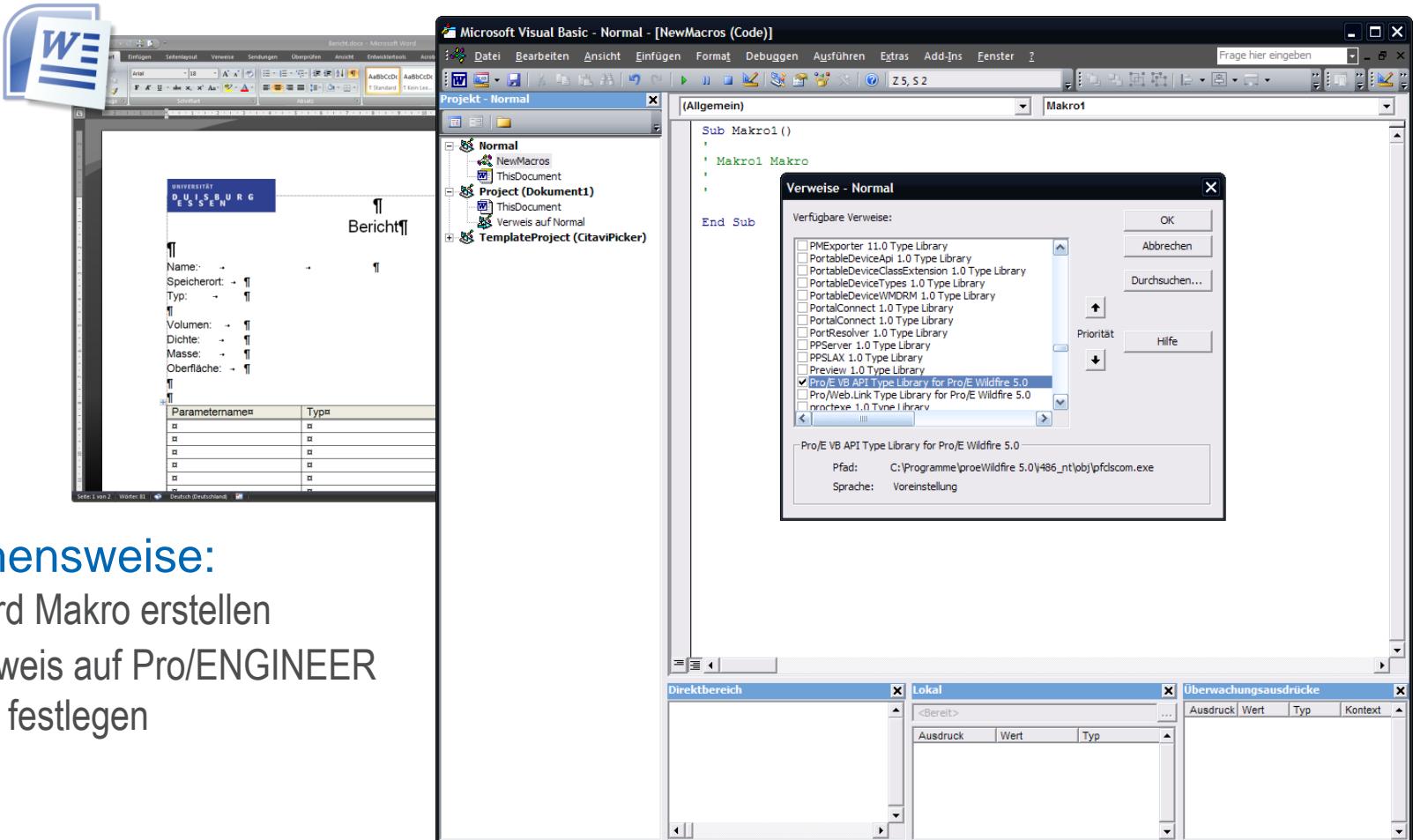


## Die Visual Basic Programmierschnittstelle – eine experimentelle Untersuchung der Möglichkeiten

- > 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 illustrates the process of generating a report in Microsoft Word using VBA. On the left, a Microsoft Word document titled "Bericht" is shown, containing a table with parameter fields and a placeholder for a part's properties. On the right, the Microsoft Visual Basic - Normal - [NewMacros (Code)] window is displayed, showing the VBA code for the macro "Makro1". The code uses the "TemplateProject (CitaviPicker)" library to interact with Pro/ENGINEER parts. A "Verweise - Normal" (References) dialog box is open, listing various type libraries, with "Pro/E VB API Type Library for Pro/E Wildfire 5.0" selected. At the bottom of the screen, several toolbars are visible, including "Direktbereich", "Lokal", and "Überwachungsausdrücke".

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

Verweise - Normal

- PMEExporter 11.0 Type Library
- PortableDeviceApi 1.0 Type Library
- PortableDeviceClassExtension 1.0 Type Library
- PortableDeviceTypes 1.0 Type Library
- PortableDeviceWMDRM 1.0 Type Library
- PortalConnect 1.0 Type Library
- PortResolver 1.0 Type Library
- PPServer 1.0 Type Library
- PPSLAX 1.0 Type Library
- Preview 1.0 Type Library
- ✓ Pro/E VB API Type Library for Pro/E Wildfire 5.0
- Pro/WebLink Type Library for Pro/E Wildfire 5.0
- nnrctext 1.0 Type Library

Pro/E VB API Type Library for Pro/E Wildfire 5.0  
Pfad: C:\Programme\proEWildfire 5.0\486\_nt\obj\pfclscm.exe  
Sprache: Voreinstellung

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

## > Vorgehensweise:

- Verbindung zu Pro/ENGINEER Sitzung
- Teil aus Verzeichnis ansprechen
  - Funktion: GetModelFromFileName(„Name“)
- 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

The screenshot shows the Microsoft Word VBA editor with a macro named 'Makro1'. The code is written in VBA and interacts with the Pro/ENGINEER API (Ipfca) to read a part file and extract specific parameters and their values.

```
Sub Makro1()
    Dim asyncConnection As Ipfca.AsyncConnection
    Dim cAC As New CCpfca.AsyncConnection
    Dim session As Ipfca.BaseSession
    Dim model As Ipfca.Model
    Dim ParaOwner As Ipfca.ParameterOwner
    Dim parameter As Ipfca.Parameter
    Dim parameters As Ipfca.Parameters
    Dim Value As Ipfca.ParamValue
    Dim solid As Ipfca.Solid
    Dim solidproperties As Ipfca.MassProperty

    Set asyncConnection = cAC.Connect(Null, Null, Null, Null)
    Set session = asyncConnection.session
    Set model = session.GetModelFromFile("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

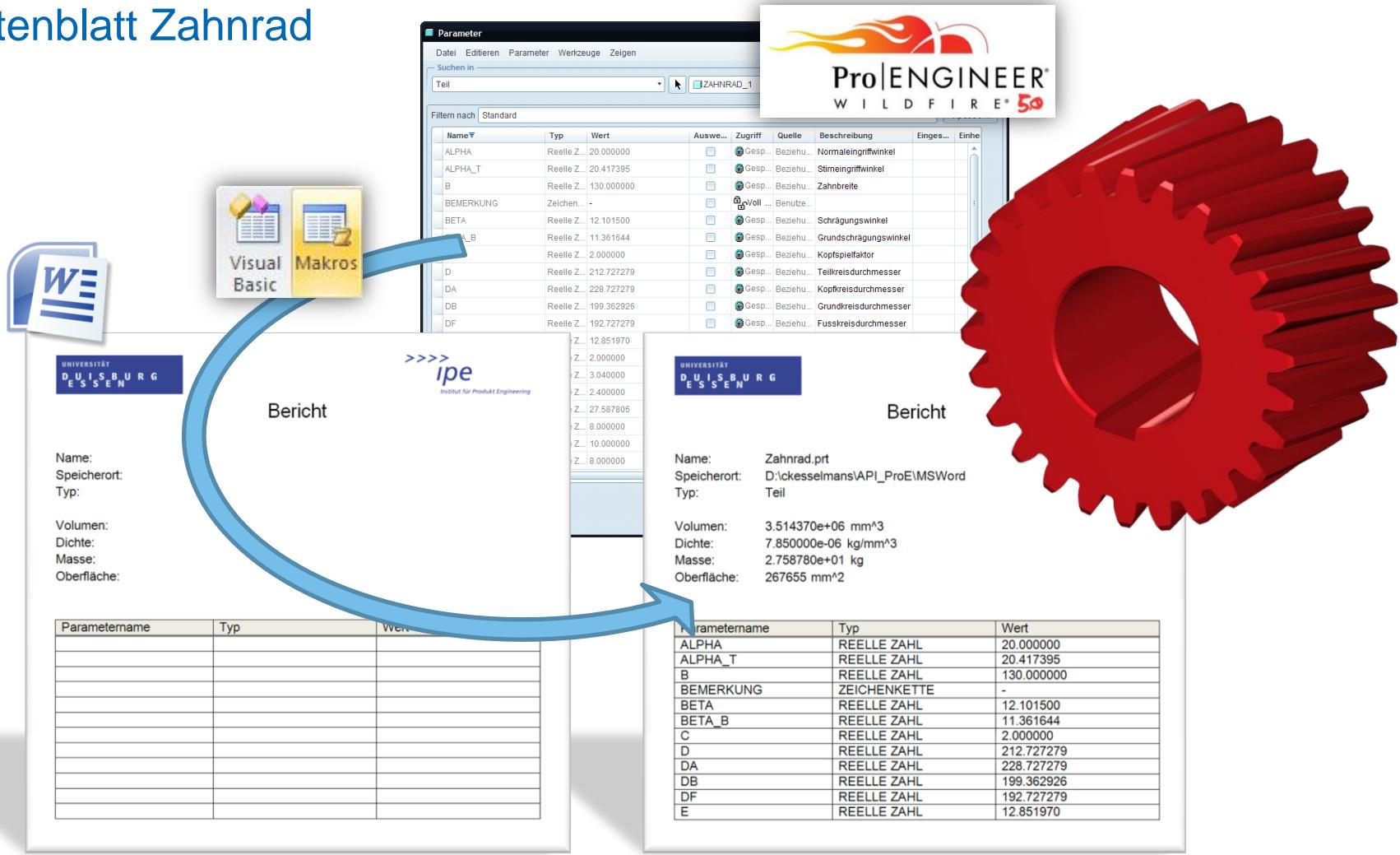
    Set solid = model
    Set solidproperties = solid.GetMassProperty(Nothing)
    Selection.TypeText Text:=CStr(solidproperties.Mass)
    Selection.TypeParagraph

End Sub
```

# Zugang über MS Office Programme (VBA)

## 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 a Microsoft Excel window titled "Tabelle\_API.xls [Kompatibilitätsmodus] - Microsoft Excel". The ribbon tabs are "Start", "Einfügen", "Seitenlayout", "Formeln", "Daten", "Überprüfen", "Ansicht", and "Entwicklertools". The "Entwicklertools" tab is selected, showing icons for "Makro aufzeichnen", "Relative Aufzeichnung", "Visual Basic", "Makrosicherheit", "Einfügen Entwurfsmodus", "Code anzeigen", "Eigenschaften zuordnen", "Importieren", "Erweiterungspakete", and "Exportieren".

A table named "Tabelle1" is visible on the left, containing the following data:

	A	B	C	D	E
1	Name	Typ	Wert		
2	Parameter_1	Reelle Zahl	100		
3	Fase	Boolean	false		
4	Parameter_2	Reelle Zahl	200		
5	Parameter_3	Reelle Zahl	200		
6	Info_1	String	Achtung		
7	Parameter_4	Reelle Zahl	30		
8	Parameter_5	Reelle Zahl	90		
9	Rundung	Boolean	false		
10	Absatz_1	Boolean	true		
11	Feature_1	Boolean	false		
12	Bearbeiter	String	Name		
13	Parameter_6	Reelle Zahl	70		
14	Material	String	Stahl		
15	Verwendung	String	sonderteil		
16	Bohrung	Boolean	false		
17	Absatz_2	Boolean	true		
18	Info_2	String	text		
19	Parameter_7	Reelle Zahl	150		
20	Parameter_8	Reelle Zahl	180		
21	Parameter_9	Reelle Zahl	20		
22	Parameter_10	Reelle Zahl	9		

The VBA editor window is open, showing a module named "makro1" with the following code:

```

Sub makro1()
    Dim asyncConnection As IpcfcAsyncConnection
    Dim cAC As New CCpfcAsyncConnection
    Dim session As IpcfcBaseSession
    Dim model As IpcfcModel
    Dim ParaOwner As IpcfcParameterOwner
    Dim Value As IpcfcParamValue
    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.GetModelFromFile("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 "String"
                Set Value = CParamvalue.CreateStringParamValue(Wert)

            Case "Boolean"
                Set Value = CParamvalue.CreateBoolParamValue(Wert)

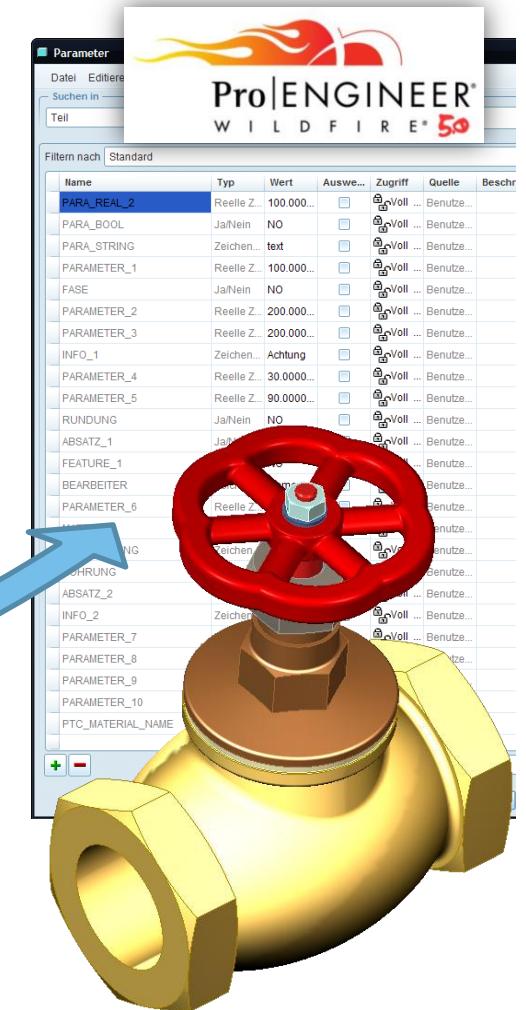
        End Select

        Set para = ParaOwner.CreateParam(Name, Value)

    Next col
End Sub

```

Two blue arrows point from the "Tabelle1" table towards the "CParamvalue" object in the VBA code.



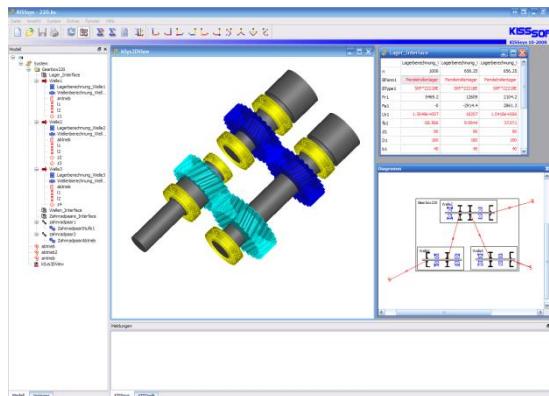
## Die Visual Basic Programmierschnittstelle – eine experimentelle Untersuchung der Möglichkeiten

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

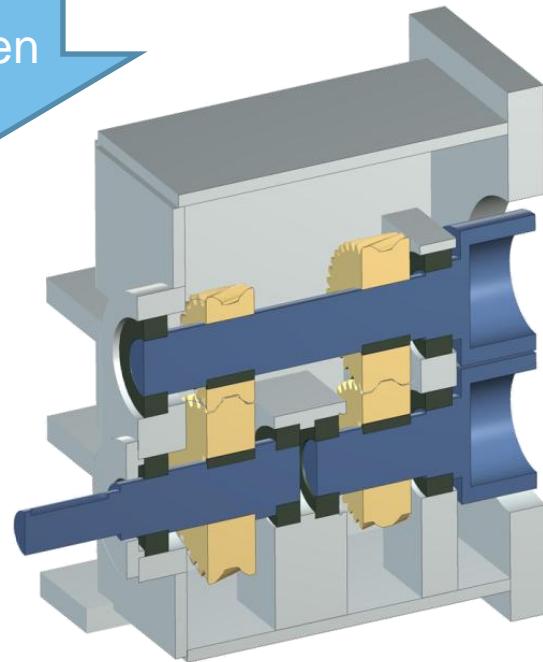
## Projekt: Automatisierte Getriebeableitung mit Anbindung an Berechnungssoftware

### > 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



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



## Ablauf einer Getriebeauslegung

1. Vorgabe der Eingangsdaten



2. Grobauslegung Zahnradparametern

Getriebe Konfigurator

Allgemeine Eingaben | 2. Mdesing-Berechnungen | 5. ProE Modell | 6. Datenspeicherung

Eingabe

Antriebsdrehzahl: 1400 min^-1  
Abtriebsdrehzahl: 980 min^-1  
Leistung: 160 kW

Soll Uebersetzung 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]

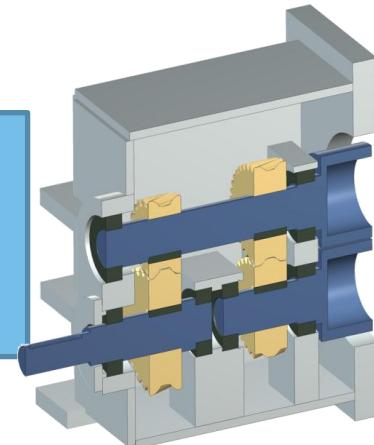
Ritzelwelle oder Welle Zahnrad Kombination  
Welle-Ritzel Kombination

Programm verlassen

3. Manipulation der XML-Dateien für die Festigkeitsnachweise

```
version="1.0" encoding="UTF-16" standalone="yes">
<?xml version="1.0" encoding="UTF-16" standalone="yes"?>
<!DOCTYPE BFMEDAT SYSTEM "BFMEDAT.DTD">
<BFMEDAT>
  <Signature>
    Version
    CreateBy
    CalcID
    CalcName
    ModuleVersion
    Header
    Data
      <Parameter>
        <Name>z1</Name>
        <Type>double</Type>
        <Value>51</Value>
        <MeasureBase>mm</MeasureBase>
      </Parameter>
      <Parameter>
        <Name>z2</Name>
        <Type>integer</Type>
        <Value>21</Value>
        <MeasureBase>mm</MeasureBase>
      </Parameter>
      <Parameter>
        <Name>b1</Name>
        <Type>double</Type>
        <Value>120</Value>
        <MeasureBase>mm</MeasureBase>
      </Parameter>
    </Data>
  </Header>
  <Calculation>
    <Zahnradsatz>
      <Zahnradpaar>
        <Zahnrad>
          <Name>z1</Name>
          <Type>Ritzel</Type>
          <Value>51</Value>
          <MeasureBase>mm</MeasureBase>
        </Zahnrad>
        <Zahnrad>
          <Name>z2</Name>
          <Type>Welle</Type>
          <Value>21</Value>
          <MeasureBase>mm</MeasureBase>
        </Zahnrad>
      </Zahnradpaar>
    </Zahnradsatz>
  </Calculation>
</BFMEDAT>
```

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



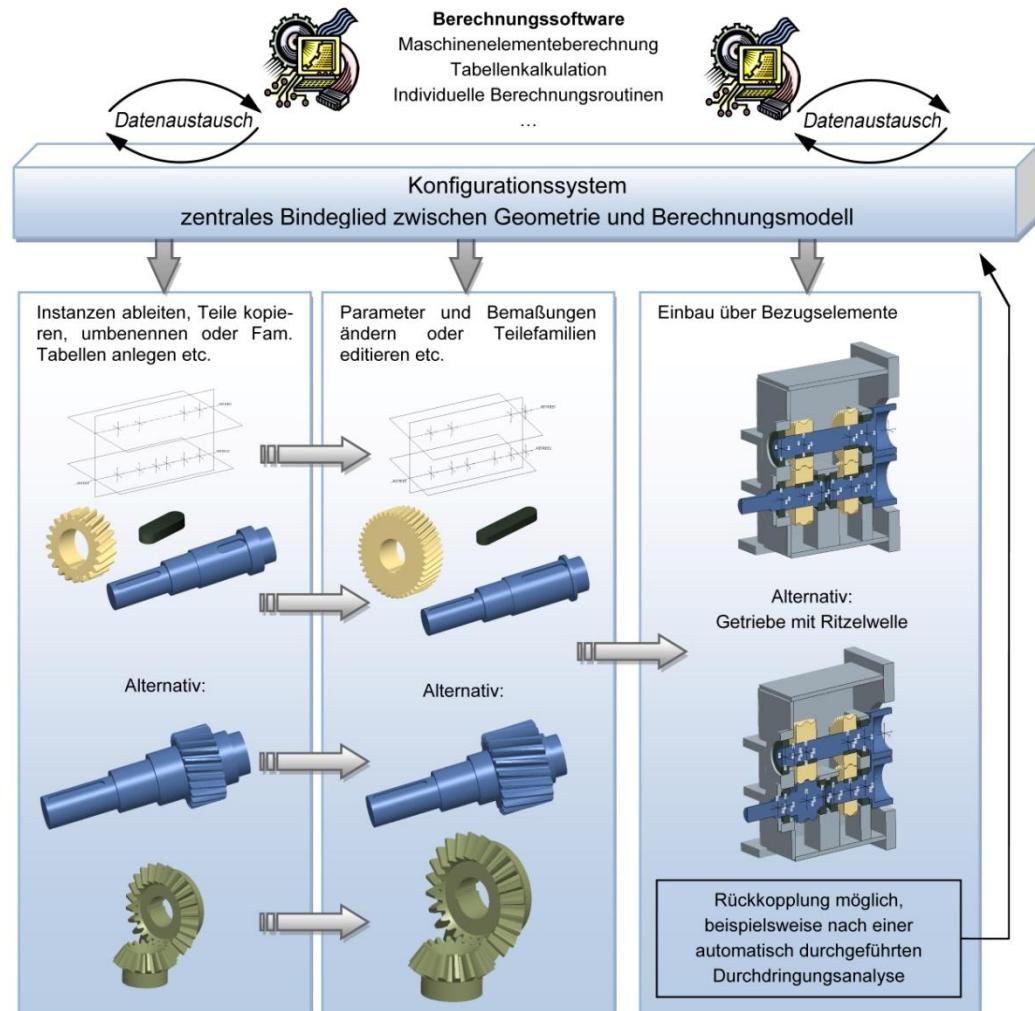
## Benötigte API-Funktionalitäten zum Aufbau eines Getriebes

### > 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

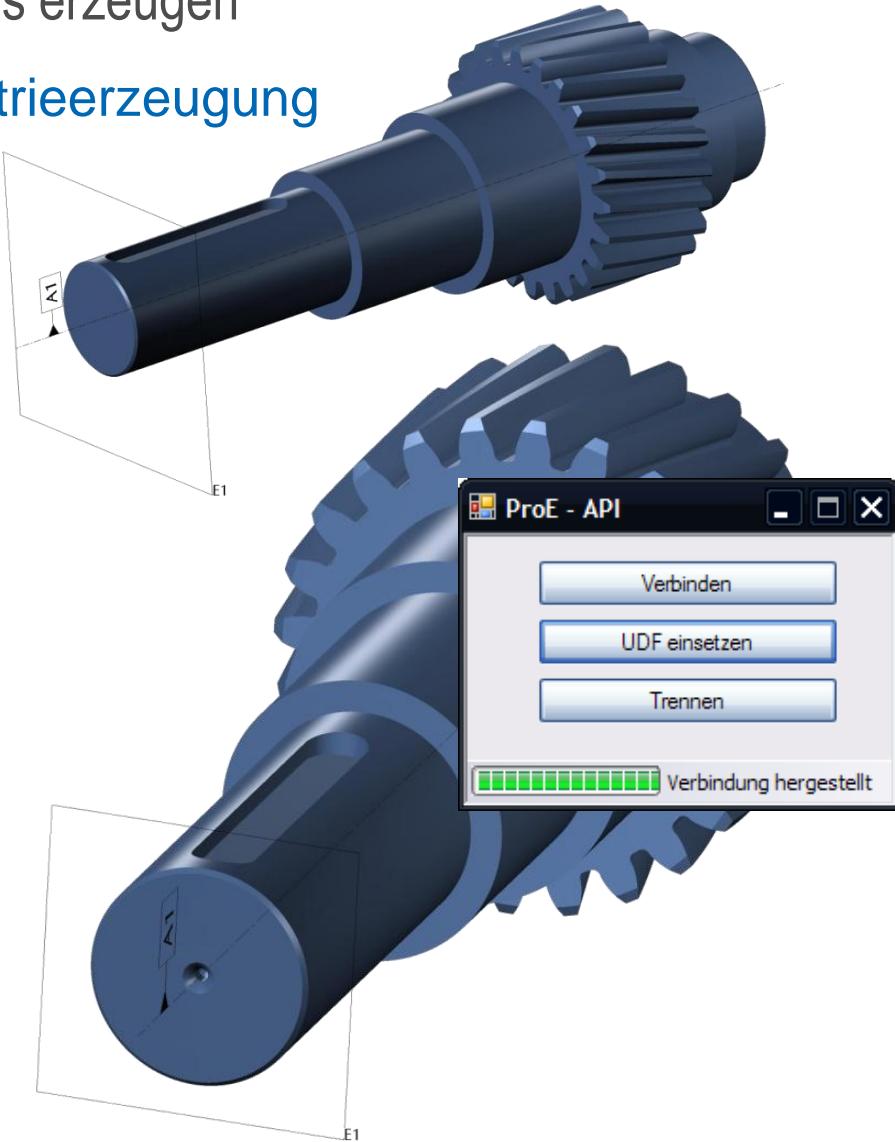
> Zugang zur automatischen Geometrieerzeugung

> Drei Möglichkeiten zur Referenzauswahl:

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

> Beispiel Zentrierbohrung:

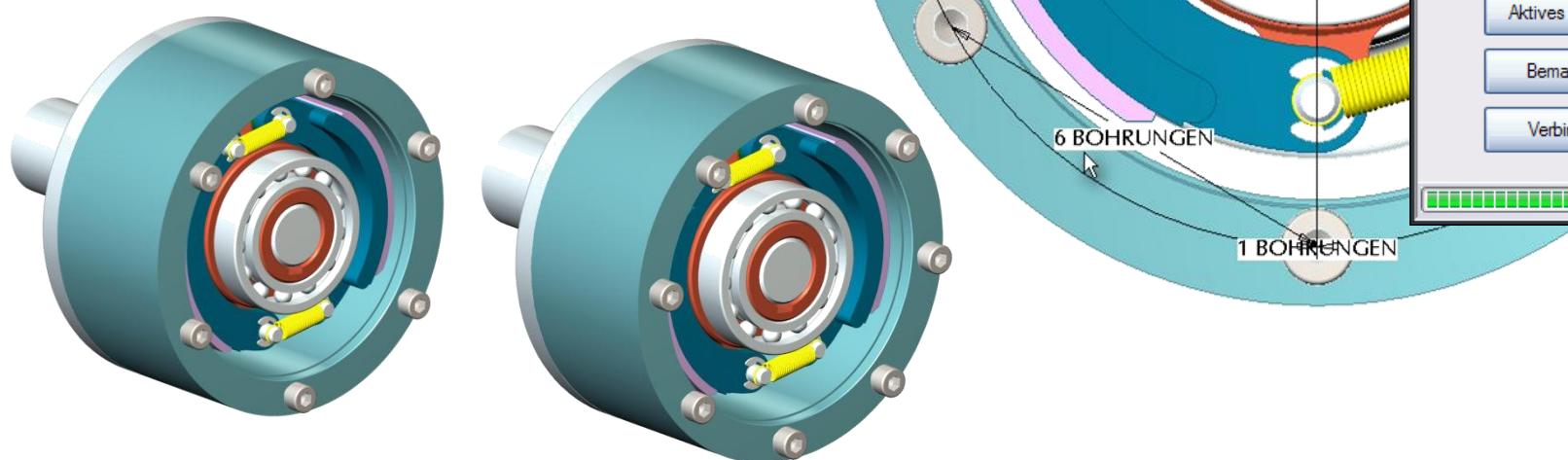
- UDF manuell erstellt
- Referenzen (E1 und A1) sind im Zielteil bekannt



## 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



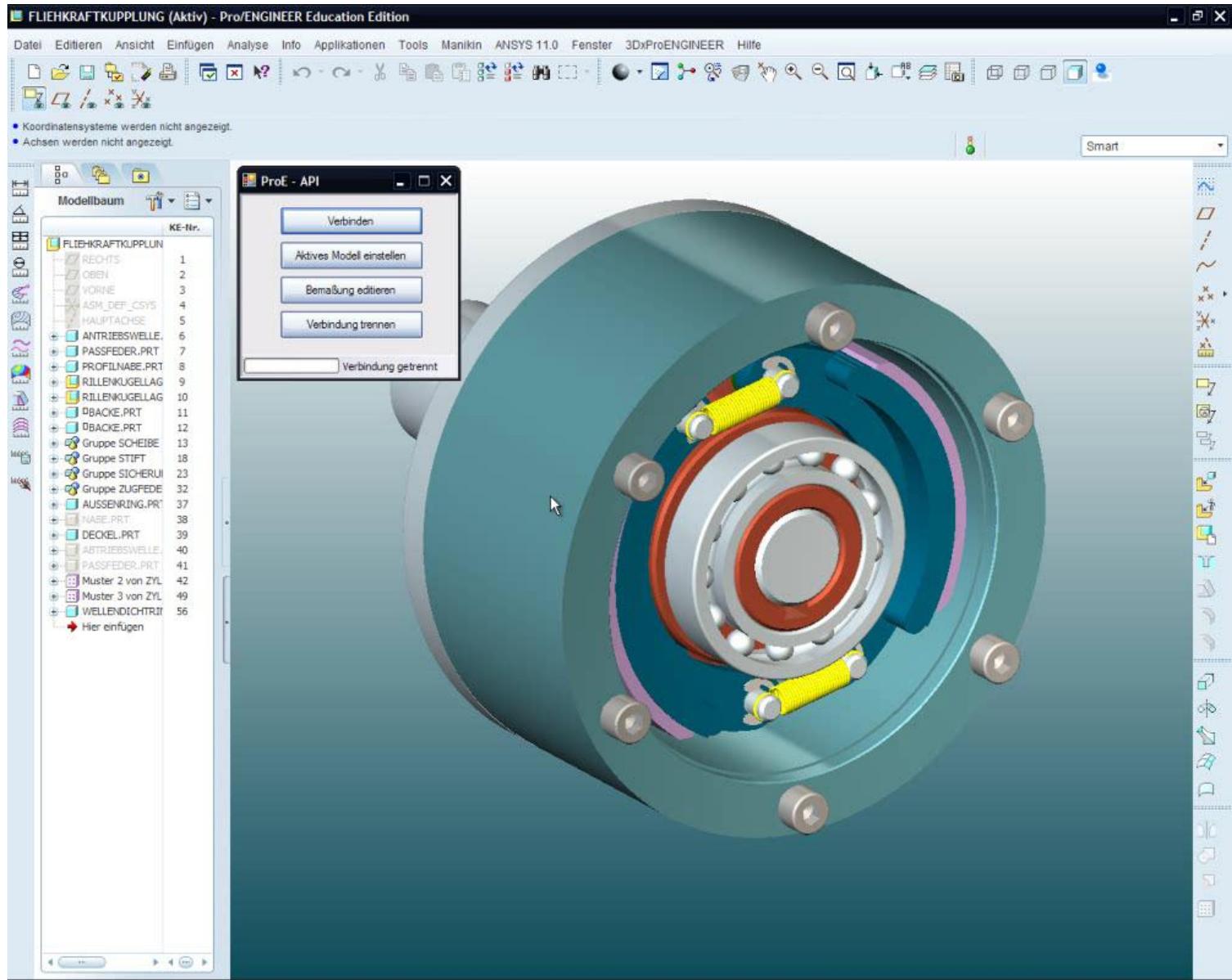
# Zugang über VB.Net Anwendungen

>>>

ipe

Institut für Produkt Engineering  
Rechnereinsatz in der Konstruktion

planet PTC  
LIVE



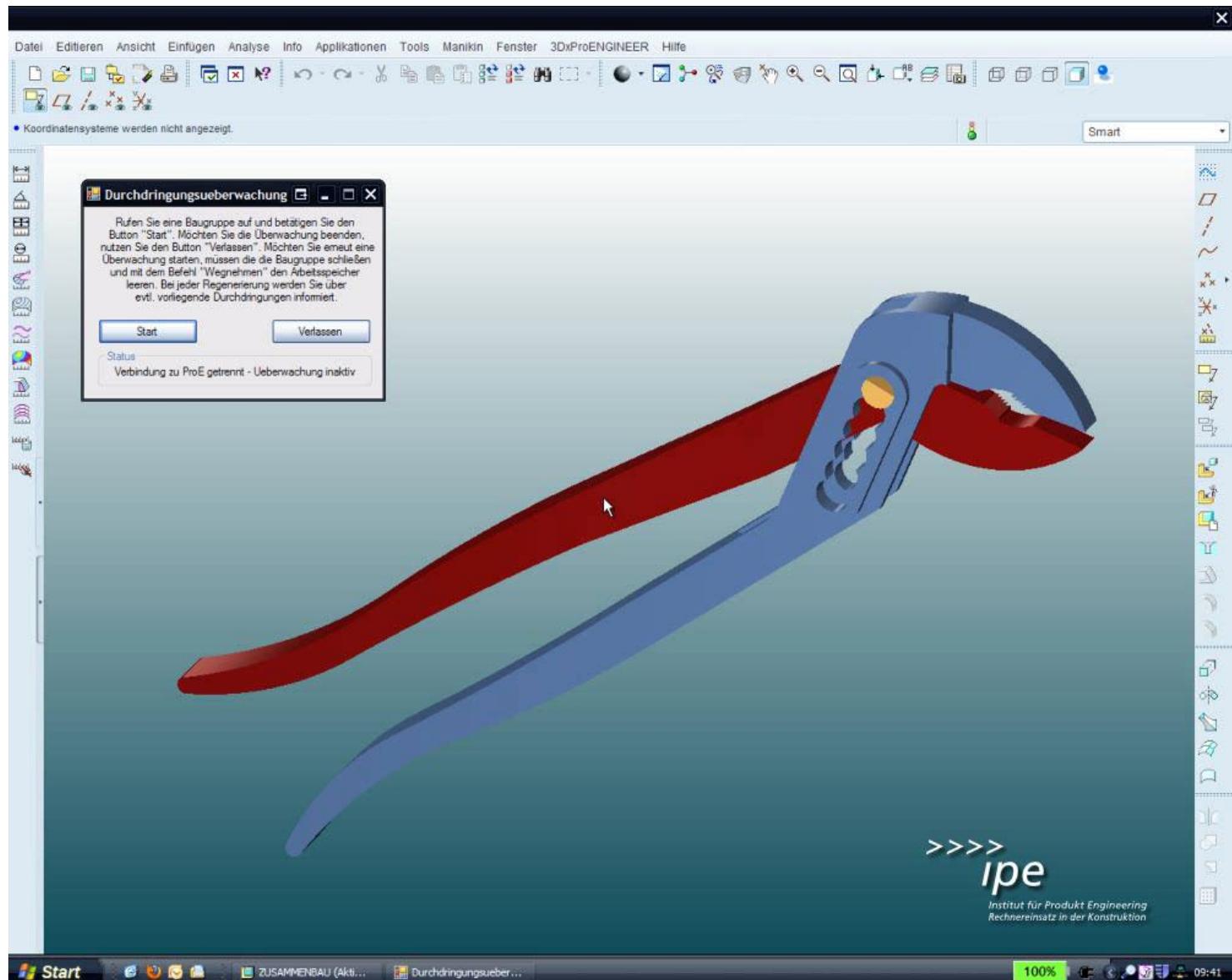
Beispiel: Auf Ereignisse reagieren - Durchdringungsüberwachung

## > 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



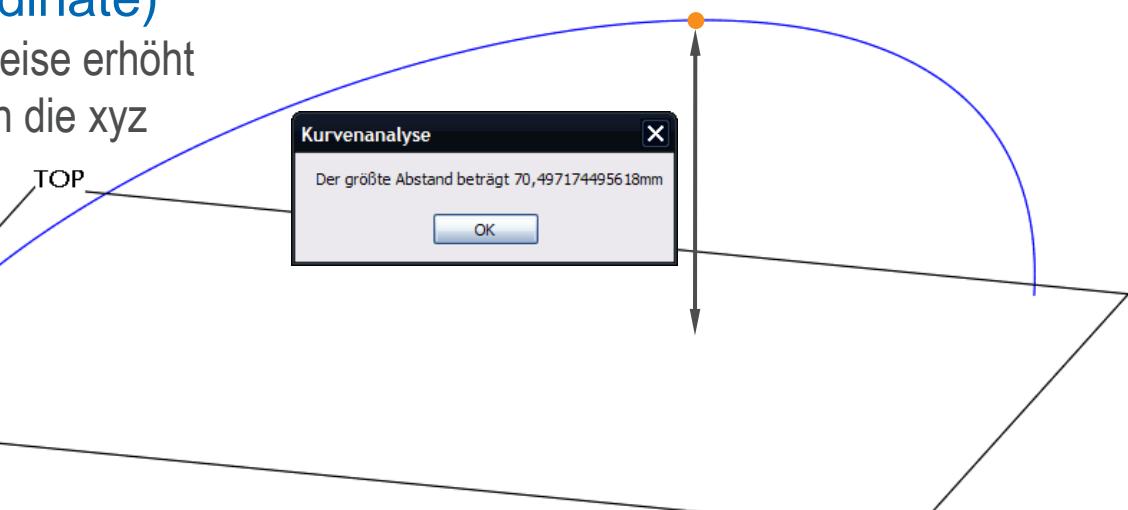
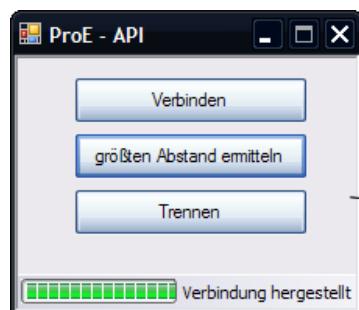
# Zugang über VB.Net Anwendungen



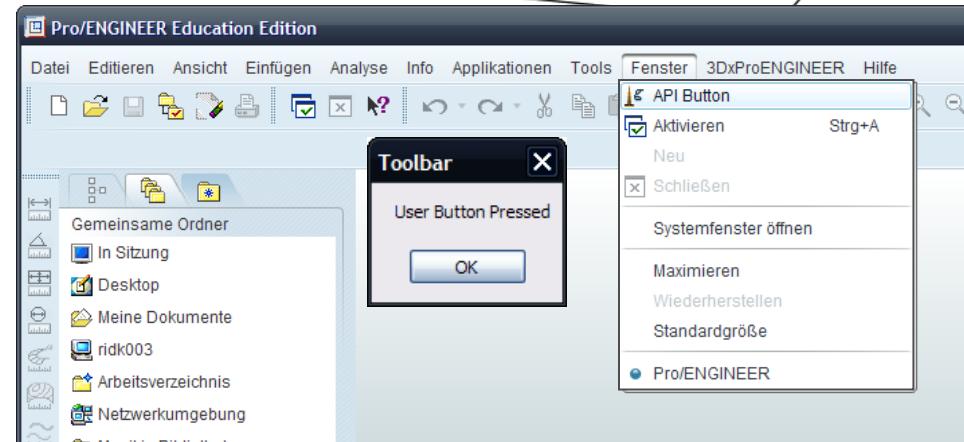
Beispiel: Kurvenanalyse und Menüeintrag

## > 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

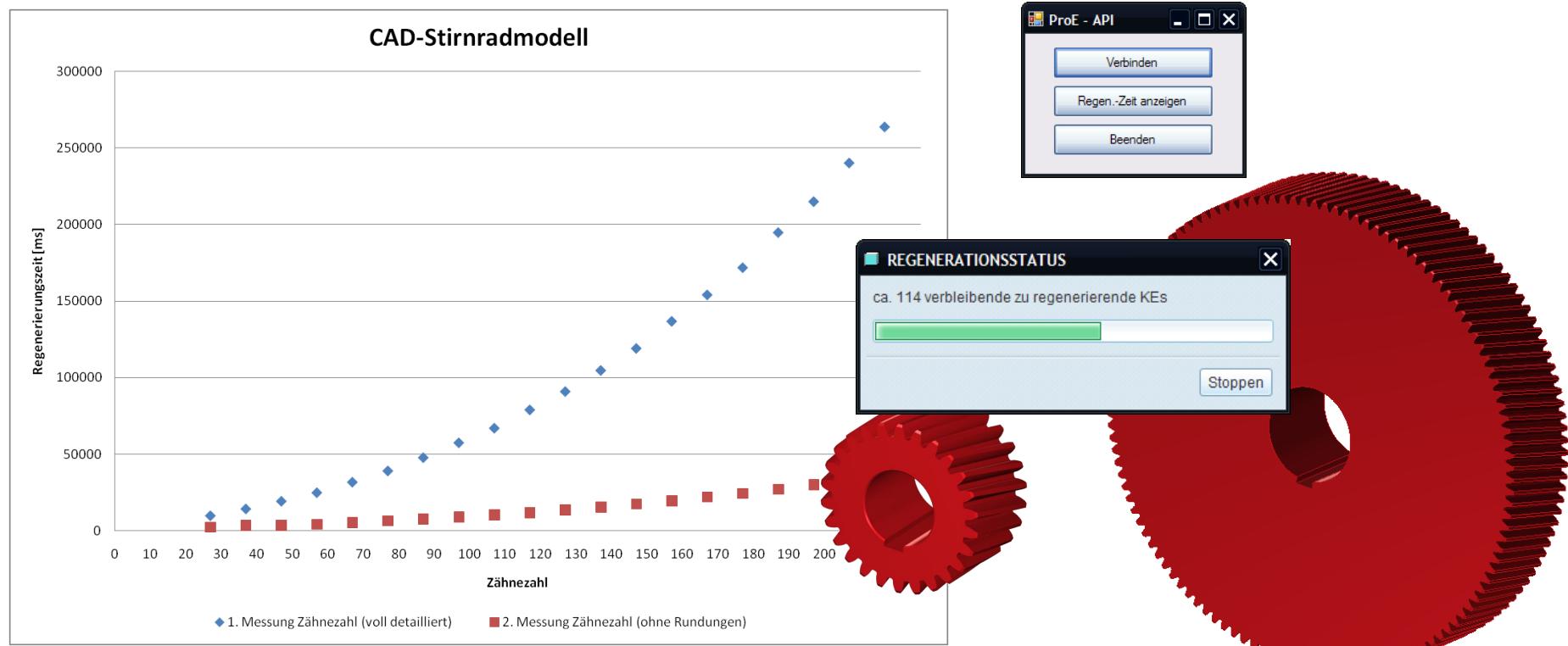


## > Menüeinträge zur Laufzeit des Programms generieren und deren Auswahl überwachen



## 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!