Automatisierte Erstellung von Ergebnisberichten

Brose Fahrzeugteile GmbH & Co. KG, Hallstadt
Thomas Sauernheimer
CADFEM GmbH
Ralph Echter
Automatic report generation

Content

- FEA at Brose Doorsystems
  - Brose product portfolio
  - CAE in Brose development process
  - Motivation for automatic report
  - Report requirements

- Project realization (CADFEM)
  - ANSYS Workbench Project Schematic customization
  - ANSYS Mechanical customization
Automatic report generation

Brose product portfolio

Structures and components for vehicle seats

Systems for engine cooling, electric motors and drives

Modules and components for vehicle doors
Automatic report generation

- CAE in Brose development process

Phases:
- Effort estimation
- Basic calculation of concepts
- Standard FEA for test release V0 and V1
- Test release V2/lessons learned
- Basic calculation optimization/quality issues

Effort estimation → Basic calculation of concepts → Standard FEA for test release V0 and V1 → Test release V2/lessons learned → Basic calculation optimization/quality issues

SOP → series → spare parts business
Automatic report generation

- **Motivation for automation**
  - About 120 window regulator calculations a year
  - About 300 reports a year
  - High potential for automation
  - Automation of model build up was done in 2012
    (ACUM 2013; Process automation with Jscript in ANSYS Workbench)
  - First automatic report was a „Quick and Dirty“ solution
  - High quality automatic report for internal customers was missing
Automatic report generation

• Report requirements
  – Use of Workbench images, figures
  – Input of project data
  – Report for single load case
  – Collected report for all required load cases
  – Editable
  – Standard layout
  – Automatic and individual caption
  – Automatic information generation (bill of materials…)
  – Input of load case evaluation
  – Sub model technology
Automatic report generation

- **Project realization**

  - Definition of report requirements
  
  - Discussion with CADFEM how to realize
  
  - Necessary programming KnowHow does not exist at simulation department BROSE Hallstadt
    - JScript ✅
    - Python 🔴
    - HTML + CSS ❌

  - CADFEM is able to combine all programming methods
    - CADFEM does the programming work
Automatic report generation

- **Report built up**

**ANSYS Project**
- Overall report
- Input CAE data
- Choose load cases

**ANSYS Mechanical**
- Postprocessing Item
- Single report
- Create Images / Figures
- Input evaluation

Diagram:
- ANSYS Project
- HTML Report
- PDF Print
- Edit PDF
- Load case evaluation
- CAE data

Verfasser: Sauernheimer, Thomas  Funktion: TVE  Date saved: 22.05.2015
Automatic generation of reports - Customization by CADFEM
Overview – Automatization in ANSYS Workbench
Description of software customization for BROSE

- ANSYS Workbench Project Schematic customization
- **Project Header Data** programming – customized menu “Brose Auto Report”
Description of software customization for BROSE

- ANSYS Workbench Project Schematic customization
- **Project Header Data** programming – customized menu “Brose Auto Report”
  - Applied languages: **Python + Windows Forms**
  - Code sequences in Python script

```python
import os
import clr
clr.AddReference("System.Windows.Forms")
clr.AddReference("System.Drawing")
from System.Windows.Forms import *

class IForm(Form):
    def __init__(self):
        Form.__init__(self)

        # Überschrift des Fensters
        self.Text = 'Project Header Data'
        self.BackColor = Color.White

        # Icon oben links im Fenster
        self.Icon = Icon("\images\process.ico")

        #-------- System 1: CAE Number --------
        label1 = Label()
        label1.Text = "CAE Number:")
        label1.Size = Size(sizeLabel, height)
        label1.Location = Point(dist_l,dist_h)
```

Importing Windows Forms capabilities to create and fill window - see previous slide with content

Specifications on window content - see previous slide e.g.
- header,
- images,
- systems details,
- …
Description of software customization for BROSE

- ANSYS Workbench Project Schematic customization
- Load case selection programming for Overall Report (HTML) – customized menu “Brose Auto Report”
- Applied languages: Python + Windows Forms
Description of software customization for BROSE

• ANSYS Workbench Project Schematic customization
  • Overall report (HTML) programming
    • Applied languages: Python + JScript + HTML + CSS
    • Workflow in Python:
      • Store connectivity of systems with global-submodel relation
      • Loop through all selected systems
        • Save user-specified data:
          • Images, material data, temperature settings, CAE information of systems with global-submodel connectivity to folders
          • Images, material data, temperature settings, CAE information of single systems to folders.
    • Folder structure:
Description of software customization for BROSE

• ANSYS Workbench Project Schematic customization
  • Overall report (HTML) programming
    • Workflow in JScript:
      • Retrieve user-specified data from ANSYS Mechanical by JScript;
      • JScript function call from Python via “SendCommand()” function;

Example Code Python Call of JScript for Single systems:

```python
# Send JScript Code to Mechanical Simulation via Python Script
myCommand = ""
of = open("BROSE_SingleModel_Data.js","r")
myCommand += f.read()
f.close()
setupSingleSystem.SendCommand(Command=myCommand)
setupSingleSystem.Exit()
```

Application of different JScript files (*.js) for global systems, submodel systems and single systems due to different data extraction
Description of software customization for BROSE

- ANSYS Workbench Project Schematic customization
- Overall report (HTML) programming
  - Workflow in HTML:
  - HTML Code written inside Python overall report file

```python
msg = ''
msg += '<html>
msg += '<head>
msg += '<style type="text/css">@n'
cssFilePath = customizationDir + '\Scripts\Brose_report.NEW.css'
cssFile = open(cssFilePath, 'r')
msg += cssFile.read()
cssFile.close()
msg += '</style>
msg += '</head>
msg += '</body>
```

HTML layout structuring via CSS (Cascading Style Sheets):

```css
.firstPage
  margin-top: 0.5cm;
padding-left: 0.5cm;
padding-right: 0.5cm;
page-break-after: always;
border: 2px solid #777777;
min-height: 26cm;
height: 28cm;
width: 19cm;

.headerimage
  text-align: right;
padding-bottom: 0.25cm
```

© CADFEM 2015  ACUM2015_Automatisierte Erstellung von Ergebnisberichten
Description of software customization for BROSE

- ANSYS Mechanical customization
- **Single report** (HTML) programming – customized toolbar “Brose Postprocessing”
- Applied languages: **Python +XML+ JScript + HTML + CSS**
- ACT Postprocessing Toolbar feature: Automatic project header data import, additional user input, report generation: **Python +XML:**
  - XML: Defines + configures content of extension,
  - Python: Respond to user / GUI interactions, implementation of behavior/ functionalities of extension;

- **XML Code Sequence:**

```xml
<extension version="1" name="BrosePF">
  <script src="/ext.Folder\main.py" />
  <interface context="Mechanical">
    <images>/ext.Folder\images</images>
    <toolbar name="BrosePF" caption="Brose Postprocessing">
      <entry name="Brose Postprocessing" icon="Brose_i">
        <callbacks>
          <onclick>createPFFeature</onclick>
        </callbacks>
      </entry>
    </toolbar>
    <callbacks>
      <onpostfinished>callJScript</onpostfinished>
    </callbacks>
  </interface>
</extension>
```

Call Python functions
Description of software customization for BROSE

• ANSYS Mechanical customization
  • **Single report** (HTML) programming – customized toolbar “Brose Postprocessing”
    • ACT Postprocessing Toolbar feature: Automatic project header data import, additional user input, report generation: **Python +XML:**
      - XML file: Defines +configures content of extension.
      - Python script: Respond to user / GUI interactions, implementation of behavior/ functionalities of extension;

• **Python** functions invoked by XML callbacks
  ```python
def createPFeature(currentAnalysis):
    load = currentAnalysis.CreateResultObject("BrosePP")

def callJScript(analysis):
  for result in analysis.ResultObjects:
    compare_header_file(result)
  installDir = ExtAPI.ExtensionManager.CurrentExtension.InstallDir
  filePath = installDir + "\\" + "SingleReport.js"
  filePath = filePath.replace("\\", "\\\")
  sCmd1 = "fso = new ActiveXObject(\"Scripting.FileSystemObject\")\n  ExtAPI.Application.ScriptByName("jscript").ExecuteCommand(sCmd1)
  ExtAPI.SelectionManager.ClearSelection()
```
  Creates “Brose Postprocessing” postprocessing object in tree

Calls JScript file “SingleReport.js” from Python for HTML report same way as for overall HTML report
Description of software customization for BROSE

• ANSYS Mechanical customization

• **Single report** (HTML) programming – customized toolbar “Brose Postprocessing”

• Applied languages: **Python + JScript + HTML + CSS**

• ACT Postprocessing feature: Automatic project header data import, additional user input, report generation:
Automatic report generation

- **Report layout**
  - Standard layout for each report
    - CAE input data
    - Evaluation summary
    - Nondisclosure contract
    - First load case, evaluation
    - First load case, images...
    - First load case, bill of materials
    - Second load case....
Automatic report generation

- Report layout
  - Sub model technology is visible
  - Linked rough model transparent behind sub model
Automatic report generation

- Automatic report build up summary