Performing Customized Post-Processing Using Design Assessment in ANSYS Mechanical
Overview

• What is Design Assessment?
• How can I use this to solve regulatory compliance?
• How can I use this to provide my own post-processing requirements, where previously I used MAPDL?
What is Design Assessment?

• Design Assessment (or DA) is an ANSYS Workbench system that provides a framework to conduct customized post-processing of ANSYS Mechanical solutions.
• Includes both direct and user customizable features.
• ANSYS are providing some licensed tools that make use of this framework, but the primary benefit is the ability for the user to define their own customized processes.
Design Assessment - Direct Capability

• Design Assessment System
  – Out of the box, Design Assessment allows the user to factor and combine results from one or more existing analyses
  – Think of this as a more flexible and powerful version of solution combinations available from Mechanical
Design Assessment Application

- Analysis types that may be processed are:
  - Static structural
  - Transient dynamics (at a specific time)
  - Modal
  - Harmonic response
  - Response spectrum
  - Explicit dynamics
• Design Assessment System
  – Advanced post-processing of FEA results
    • Targeted at user wanting to do design evaluation based on FEA models and results
  – Predefined scripts supplied for ANSYS supplied post-processing tools
  – Enables the customer to define additional data that is associated with their model and then perform custom post-processing
    • Custom definition of input data
    • Custom result definitions
    • Custom Solve & Post scripts (Python based)
Design Assessment
- Components of the System

• On the previous slide we referred to Python based scripts
• What are these, and what other components do we need to enable the Design Assessment process?
  – Python is an interpreted, high level programming language. Incorporates the power of traditional programming languages, such as C++, but with clearer syntax, and does not require a compiler
  – Also use XML, which is a textual based data format often used for representing arbitrary data structures
So what do they do for the user?

- Think of the combination of the two as a modern way to undertake similar tasks as is achieved through APDL scripting
- XML provides the user interface data requirements i.e. what data is required to be put in, and what results are needed to be presented
- The python scripts define the way this data is to be used, either directly or by accessing other processes
Design Assessment - Basic Process

• Each Design Assessment system consists of three stages, similar to that associated with the Mechanical Systems
• First we must choose which type of assessment we want to create
  – Combination only
  – Beamcheck (discussed later)
  – Fatjack (discussed later)
  – User defined (discussed later)
Design Assessment - Basic Process

• Having chosen the type of assessment we want to undertake, we now need to go through the stages of the process
  – Setup
  – Solution
  – Results
Design Assessment - Model Setup

• Design Assessment, Setup

• Define load case combination
• Select additional “Attributes” and define associated data
  – Required for advance post-processing
Design Assessment - Solution Definition

- Design Assessment, Solution
  - Perform load case combination
    or
  - External script to perform load case combinations
  - Carry out additional “post-process” of results if required
  - Model and FEA results recovered from Mechanical system and attribute information read from Design Assessment
Design Assessment - Post-Processing

• Design Assessment, Results
  • Display standard results from a load case combination
    or
  • Pass results from external post-processing scripts back to display in Design Assessment
Load Case Combinations
Load Case Combinations

- Available from all Structural Mechanical products
- Is a more flexible incarnation of solution combination, supporting both static and transient dynamic solutions
- Provides following standard Mechanical results
  - Stress Tool
  - Fatigue Tool
  - Contact Tool (for the following contact results: Frictional Stress, Penetration, Pressure, and Sliding Distance)
  - Beam Tool
  - Beam Results
  - Stresses
  - Elastic Strains
  - Deformations
Load Case Combinations
Regulatory Compliance
DA for Regulatory Compliance

- Many industries require that designs satisfy prescribed Regulatory Compliance
  - Nuclear
  - Civil
  - Oil & Gas

- Design Assessment provides the framework for developing customized post-processing capabilities within the ANSYS Workbench environment that would enable the Regulatory Compliance

- ANSYS are providing additional licensable tools accessible from within Design Assessment that enable the checking for compliance to be automated for lattice structures in the Civil and Oil & Gas industries
  - “Code Checking” for yield, buckling, joint utilization. Consider as the inverse of the safety factor
  - Joint fatigue
Design Assessment
- Additional Licensed Products

- **ANSYS FATJACK**
  - Beam joint fatigue assessment
  - XML and Python scripts provided

- **ANSYS BEAMCHECK**
  - Joint and member utilizations
  - XML and Python scripts provided
Design Assessment - Attribute Selection

- When Beamcheck is requested the available attribute groups become visible and data selections can be made and defined.
Custom Results Processing
Customization

• This is where the real power of the Design Assessment system is demonstrated
• Provides an environment to access any external process to manipulate results from the ANSYS Mechanical analysis and present the results back in ANSYS Mechanical
• Examples include
  – Accessing APDL scripts
  – Input/output to spreadsheets
  – Integration of third party software
Customization Components

• There are three basic components to enable customization
  – XML file describing attributes for the custom system
    • Input parameters
    • Results parameters
    • References to external scripts for solution and evaluation
  – Solution Python script
  – Evaluation Python script
The XML File

- Consists of a number of tagged items that describe the data structure for input, output and process
- Here we are defining an input parameter with the name Limiting Stress, and is a real number (Double)
- Application provides access to a geometry selection cell so that the user can chose vertices, lines, etc. Here we only allow all the geometry to be processed
- The Validation provides limiting values for the parameter, in this case between 1 and 1e9
- The ObjId = 101 is the tag that will be referenced in a later item

```
<Attributes ObjId="2" Type="CAERepBase" Ver="2">
  <DAAttribute ObjId="101" Type="DAAttribute" Ver="2">
    <AttributeName PropType="string">Limiting Stress</AttributeName>
    <AttributeType PropType="string">Double</AttributeType>
    <Application PropType="string">All</Application>
    <Validation PropType="vector&lt;string>">1,1000000000</Validation>
  </DAAttribute>
</Attributes>
```
The XML File Continued

- To define how the input parameters are to be presented we use a DAAttributeGroup definition.
- The GroupType provides the main title that will appear in the details panel when the Design Assessment attribute group is selected. Each GroupType may have one or more SubTypes.
- The PropType references a previously defined tag, in this case 101, which we saw on the previous slide. If there is more than one input parameter to be provided then this would have a list of tags.

```
<DAAttributeGroup ObjId="100001" Type="DAAttributeGroup" Ver="2">
  <GroupType PropType="string">Limiting Stress Selection</GroupType>
  <GroupSubtype PropType="string">Stress Definition</GroupSubtype>
  <AttributeIDs PropType="vector<unsigned int>">101</AttributeIDs>
</DAAttributeGroup>
```
• Having defined our input parameters, we now need to define how to process this information.

• We make reference to two Python scripts for solution and evaluation.

• DAData and CombResults enable/disable access to DA results and Mechanical results.

```xml
<DAScripts ObjId="4" Type="DAScripts" Ver="2">
  <Solve PropType="string">C:\ANSYS\DesignAssessment_AskTheExpert\SolveStress.py</Solve>
  <Evaluate PropType="string">C:\ANSYS\DesignAssessment_AskTheExpert\EvaluateStress.py</Evaluate>
  <DAData PropType="int">1</DAData>
  <CombResults PropType="int">1</CombResults>
  <SelectionExtra PropType="vector<string>" />
</DAScripts>
```
Finally we define what results can be presented back in Design Assessment after the solution.

As with the input parameters, we define the GroupType that provides the main title that will appear in the details panel when the DA Result attribute group is selected. Each GroupType may have one or more SubTypes.

The AttributeIDs is a list of one or more results parameters previously defined.

DisplayType defines this as an element based result.

```
<DAResult ObjId="110001" Type="DAResult" Ver="2">
  <GroupType PropType="string">Element Results</GroupType>
  <GroupSubtype PropType="string">Safety Factor</GroupSubtype>
  <AttributeIDs PropType="vector<unsigned int>">1101</AttributeIDs>
  <DisplayType PropType="string">ElemCont</DisplayType>
</DAResult>
```
Solution Python Script

- Python Scripting provides the control for passing data between DA and an external solution system
- Here we are calling MAPDL with a user defined APDL macro, SafetyTool.mac

```python
def runStressSolve(DesignAssessment):
    WriteLine(DesignAssessment, RunMapdlFile, "/batch")
    WriteLine(DesignAssessment, RunMapdlFile, "/post1")
    WriteLine(DesignAssessment, RunMapdlFile, "FILE," + RESULTFILENAME)
    WriteLine(DesignAssessment, RunMapdlFile, "*USE,SafetyTool.mac," + YIELD)
    WriteLine(DesignAssessment, RunMapdlFile, "fini")
    WriteLine(DesignAssessment, RunMapdlFile, "/exit")

    RunMapdlFile.close()

    exelocation = "C:\\Program Files\\ANSYS Inc\\v130\\ansys\\bin\\intel\ansys130.exe"
    commandlinestring = " -b nolist -i RunMapdlFile.inp -o out.lis"
    proc = subprocess.Popen(commandlinestring, shell=True, executable=exelocation)
    rc = proc.wait()

    File = open("out.lis","r")
    DesignAssessment.getHelper().WriteToLog(File.read())
    File.close()

    os.chdir(originaldir)
```
Data is passed from the Design Assessment system to the APDL macro through the arguments list.

This script processes the data and then writes out a text file with the required results, in this case a safety factor against a limiting stress value.

curre = 0
*do,i,1,ecount
  curre = ELNEXT(curre)  ! get the element number
  output(i,1) = curre
  *get,output(i,2),etab,1,elem,curre
  *get,output(i,3),etab,2,elem,curre
  output(i,3) = 1/(Factor*output(i,3))
*enddo

! save the output to a file
*cfopen,results,txt
!*vwrite,
!('Element Number, Critical Layer, Safety Factor')
!*vwrite,output(1,1), output(1,2), output(1,3)
(F10.0, F8.4, F12.4)
*cfclose
Evaluation Python Script

• Once the external process has been completed we then need to evaluate the required results to be presented

• This script reads an external file created by the APDL macro, and stores the element results in the Design Assessment results

```python
# Loop over each element to store the required result
for ElementIter in range(Ecount):
    Element = Mesh.Element(ElementIter).ID()
    ResultValue = ResultStructure.ElementResultValue(Element-1)

    # Find this element in the MAPDL list
    ipos = findInIntList(ElemNumber,Element)
    if ipos >= 0 :
        print "Found element ",Element," in MAPDL results"
        if Attrib == "Critical Layer" :
            ResultValue.setValue(ElemLayer[ipos])
        elif Attrib == "Safety Factor" :
            if ElemSafety[ipos] > 10 :
                ElemSafety[ipos] = 10
            ResultValue.setValue(ElemSafety[ipos])
        else:
            print "Element ",Element," not found in MAPDL results"
            ResultValue.setValue(1.7976931348623157e+308)
        else:
            print "Element ",Element," not found in MAPDL results"
            ResultValue.setValue(1.7976931348623157e+308)
```
The Finished Product
Design assessment is fully documented in the ANSYS help system including how to:

- Perform basic analysis
- XML formatting
- Solution scripts
- API to pull/push information from Mechanical
Example Applications

- Non standard load combinations - SRSS
- APDL macro
- Fragmentation using Explicit Dynamics
- Element numbering
SRSS Result Combinations

- Standard solution combinations only allow simple linear summation
• Design Assessment provides the environment to combine solutions in whatever method desired
• Here we want to undertake a SRSS combination using a customized script
SRSS Result Combination

- Simply add the Design Assessment system to the series of existing structural systems
SRSS Result Combination

• The user interface, defined using an xml file, provides for various standard and user defined results
• First define contributing solution conditions, as before
• Then select the type of result desired (in this case the SX stress component)
The XML definition provides for either stress results, or “generic” results using expressions.

Available expressions can be found by clicking on the Worksheet button when a Solution item in the tree is selected.
SRSS Result Combination

- Include displacements, stress values and expressions
- Note the definition of units to present results in current units set
Access to MAPDL macro

- We will now look at an example of using Design Assessment to call an APDL macro to compute laminate failure safety factors, and which layer the minimum occurs.
Access to MAPDL macro

```
*dim,output,arra,ecount,3
  ! number of elements x 3

etab,layer,FCMX,lay
etab,max_SR,FCMX,val

curre = 0
*do,i,1,ecount
  curre = ELNEXT(curre
  output(i,1) = curre
  *get,output(i,2),etab,1,elem,curre
  *get,output(i,3),etab,2,elem,curre
  output(i,3) = 1/(Factor*output(i,3))
*enddo

! cap the max SF as desired
*if,LimitVal,gt,0,then
  *do,i,1,ecount
    *if,output(i,3),gt,LimitVal,then
      output(i,3) = LimitVal
    *endif
  *endif
*enddo
*endif
```
Example of composite failure analysis using MAPDL /POST1

Composite analysis (using command snippets)

DA System using /POST1 failure theories

Set Failure Props for MAPDL

Results brought back from MAPDL
Fragmentation

- AUTODYN can report fragmentation information
- This is not currently available through Explicit STR
- Can be generated using a Design Assessment script
Fragmentation

- Aluminum ring subject to large internal pressure
Fragmentation

- Ring breaks into eight fragments
- Use Design Assessment to compute fundamental properties of fragments
Fragmentation

Fragmentation Information Summary

Ident name: admodel
Number of fragments: 8
Unit system: [mm, mg, ma]
Time: 4.000E+00
Cycle: 1360
Total Mass: 1.575E+09

<table>
<thead>
<tr>
<th>Fragment Number</th>
<th>Elements in Fragment</th>
<th>Mass</th>
<th>Center X</th>
<th>Center Y</th>
<th>Center Z</th>
<th>Origin X</th>
<th>Origin Y</th>
<th>Origin Z</th>
<th>Volume</th>
<th>Characteristic Length</th>
<th>Kinetic Energy</th>
<th>Average Speed</th>
<th>2-Wire Momentum</th>
<th>3-Wire Momentum</th>
<th>Angle (EFD)</th>
<th>Angle (Dial)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>256</td>
<td>2.437E+08</td>
<td>257.6</td>
<td>529.45</td>
<td>250.00</td>
<td>191.68</td>
<td>446.08</td>
<td>250.00</td>
<td>0.00072E-07</td>
<td>1253.32</td>
<td>5.752E-12</td>
<td>216.70</td>
<td>1.272E+10</td>
<td>1.083E+04</td>
<td>0.181</td>
<td>0.000</td>
</tr>
<tr>
<td>2</td>
<td>335</td>
<td>2.044E+08</td>
<td>300.19</td>
<td>604.75</td>
<td>250.00</td>
<td>303.38</td>
<td>303.38</td>
<td>250.00</td>
<td>0.00725E-07</td>
<td>1253.77</td>
<td>5.620E-12</td>
<td>371.11</td>
<td>1.433E+10</td>
<td>1.741E+04</td>
<td>0.178</td>
<td>0.000</td>
</tr>
<tr>
<td>3</td>
<td>184</td>
<td>2.989E+08</td>
<td>194.2</td>
<td>688.38</td>
<td>250.00</td>
<td>507.50</td>
<td>507.50</td>
<td>250.00</td>
<td>0.00046E-07</td>
<td>1256.96</td>
<td>2.303E-12</td>
<td>221.13</td>
<td>3.172E+10</td>
<td>1.398E+04</td>
<td>0.192</td>
<td>0.000</td>
</tr>
<tr>
<td>4</td>
<td>150</td>
<td>1.791E+08</td>
<td>648.23</td>
<td>549.44</td>
<td>250.00</td>
<td>781.14</td>
<td>445.11</td>
<td>250.00</td>
<td>0.00036E-07</td>
<td>1032.65</td>
<td>4.248E-12</td>
<td>225.46</td>
<td>2.444E+10</td>
<td>2.422E+04</td>
<td>0.175</td>
<td>0.000</td>
</tr>
<tr>
<td>5</td>
<td>150</td>
<td>1.817E+08</td>
<td>804.11</td>
<td>714.62</td>
<td>250.00</td>
<td>793.94</td>
<td>543.14</td>
<td>250.00</td>
<td>0.00031E-07</td>
<td>817.61</td>
<td>2.303E-12</td>
<td>222.26</td>
<td>2.385E+10</td>
<td>2.362E+04</td>
<td>0.179</td>
<td>0.000</td>
</tr>
<tr>
<td>6</td>
<td>150</td>
<td>1.692E+08</td>
<td>241.5</td>
<td>1100.18</td>
<td>250.00</td>
<td>182.29</td>
<td>615.12</td>
<td>250.00</td>
<td>0.00093E-07</td>
<td>565.50</td>
<td>4.248E-12</td>
<td>223.35</td>
<td>7.102E+09</td>
<td>3.656E+09</td>
<td>0.178</td>
<td>0.000</td>
</tr>
<tr>
<td>7</td>
<td>150</td>
<td>1.899E+08</td>
<td>1903.96</td>
<td>58.31</td>
<td>250.00</td>
<td>688.01</td>
<td>44.10</td>
<td>250.00</td>
<td>0.00090E-07</td>
<td>883.75</td>
<td>4.248E-12</td>
<td>222.50</td>
<td>2.703E+10</td>
<td>1.307E+09</td>
<td>0.179</td>
<td>0.000</td>
</tr>
<tr>
<td>8</td>
<td>125</td>
<td>1.293E+08</td>
<td>405.77</td>
<td>1096.07</td>
<td>250.00</td>
<td>304.06</td>
<td>421.08</td>
<td>250.00</td>
<td>0.00536E-07</td>
<td>609.34</td>
<td>3.177E-12</td>
<td>226.80</td>
<td>1.916E+10</td>
<td>2.584E+10</td>
<td>0.175</td>
<td>0.000</td>
</tr>
</tbody>
</table>

Small Particle summary

<table>
<thead>
<tr>
<th>Total Number</th>
<th>Total Mass</th>
<th>Largest Mass</th>
<th>Average Mass</th>
<th>Total Area</th>
<th>Total Momentum</th>
</tr>
</thead>
<tbody>
<tr>
<td>256</td>
<td>8.94E+07</td>
<td>1.56E+05</td>
<td>4.51E+05</td>
<td>8.12E+05</td>
<td>3.50E+12</td>
</tr>
</tbody>
</table>
Element numbering

• Have you ever wanted to know the finite element numbers used in an analysis?
• This is a light-hearted use of Design Assessment to extract and present element numbers
Element numbering

- Design Assessment supports different scoping methods. Here we allow either geometry or named selections.
- Geometry selection uses standard WB tools for faces/edges/bodies, etc.
Element numbering

- Can get a contour plot of element numbers, but more useful to use the probe feature
Summary

• Design Assessment is a valuable extension to the Mechanical environment, providing a customization toolset for results extraction and post-processing

• Supports majority of Mechanical capabilities
  – Standard results types such as stress tool, fatigue tool, beam tool, etc
  – Units
  – Nodal, elemental nodal and element results
  – Scalar, vector and tensor results display
  – Expressions, including mathematical operators
  – Probes
Summary

• More importantly it extends the functionality to allow custom scripts to be developed to undertake complex post-processing activities
• A modern version of the APDL post-processing capability
• Several custom scripts have been created by ANSYS to undertake beam code checking and fatigue computations