Design Assessment

Eric Miller
Principal
Director, Simulation and Business Technologies

5/19/2011
PADT, Inc.
Agenda

• Note: This presentation is being recorded

• Introduction
• Overview
• Result Combinations
• The API
  – XML
  – Python Scripts
• Conclusion
Introductions
Upcoming Webinars

- May 26, 2011  12:00 pm
  Using APDL Snippets in ANSYS Mechanical

- Webinars go on Summer Breaks in June and July

- See upcoming and past webinars at:
  - padtincevents.webex.com
    - Click on ANSYS Webinar Series
About PADT

- PADT is an Engineering Services Company
  - Mechanical Engineering
  - 17 Years of Growth and Happy customers
  - 68 Employees

- 3 Business Areas
  - CAE Sales & Services
    - Consulting, Training, Sales, Support
  - Product Development
  - Rapid Prototyping & Manufacturing

- Learn More: www.PADTINC.com

“We Bring Dimension to Your Ideas”
Cube HVPC Systems

• Balance between speed and cost
  – **Mini-Cluster**
    96 Cores / 256 GB RAM / 3.6 TB Disk
    Mobile Rack / UPS / Monitor / Keyboard
    $43,250
  – **Compute Server**
    32 Cores / 128 GB RAM / 3 TB Disk
    $12,300
  – **Simulation Workstation**
    12 Cores / 64 GB RAM / 1.5 TB Disk
    $5,800
  – **Simulation Fileserver**
    10 TB Disk / External eSATA
    $5,800

• [www.CUBE-HVPC.com](http://www.CUBE-HVPC.com)
Overview

What is Design Assessment and What Can it Do?
Design Assessment

• First off – why so many S’s in Assessment!
• Design Assessment (DA) is a new system in ANSYS Mechanical R13 that allows do sophisticated post processing
• It lets you do two things
  – Solution Combination
  – Make a DA object and do an “Assessment” on your results using a predefined or user defined set of operations
• The tool lets you
  – Grab results, do operations with python, export them, run an external program, import modified results
• API is documented to allow users to make their own
  – Most powerful aspect!
Design Assessment, Cont...

- Described in the documentation as a pretty complicated and abstract capability
  - Utilizes object oriented way that ANSYS Mechanical and projects are structured

- Basic Solution Combination in ANSYS Mechanical uses API behind the scenes to do load case combinations
  - Great way to get familiar with the concept
  - Works with static structural and transient dynamics
  - Add to your model in ANSYS Mechanical

- Assessments are added as a DA system in the project schematic

- The two pre-done assessment tools are for ASUS
  - Off shore code checking: FATJACK and BEAMCHECK
  - Requires license
• Note how the Solution Combination has no analysis settings or solution selection, just results
Solution Combination
Solution Combination Process

1. Define a model with multiple:
   – Load steps, time steps, or multiple systems (but same model)
2. Insert Solution Combination object
3. Use Worksheet to specify the combination you want
4. Add results you want to look at
5. Update results and view
6. Repeat for different combinations
Our Sample Model

- **Valveypoo**
  - Fake aerospace valve
  - Fixed on either end
  - Four load steps
    1. Internal pressure of 1
    2. X acceleration of 1
    3. Y acceleration of 1
    4. Z acceleration of 1
Insert Solution Combination

- In ANSYS Mechanical
  - Select your Model
  - RMB->Insert->Solution Combination
Define Combination

• After inserting a blank worksheet comes up
• Coefficient: Scale factor to be applied to the result value
• Environment Name: Name of environment to pull results from
  – For when you have multiple systems
• Time:
  – Specify time to pull from
• RMB on it and select Add
Define Combination

• For example the results are scaled for:
  – 534 psi internal pressure
  – Acceleration vector of (10, 4.6, 2.5)
Add Result Objects

• Same as you would in your Solution branch
  – RMB->Insert or Insert Icons
• Note: Derived Values (Sig1, SigE) are combined by component and then derived.
• Detail view is just like solution branch
  – You can specify the type and the direction
  – Specify a Coordinate System
  – Decide if you want Averaged or Unaveraged, nodal diff… etc…
  – Scoping works as well
    • Geometry, path, surface, named selection, etc..
• Use RMB->Rename Based on Definition
  – In fact, naming can be very important to keep your combinations straight
View Results

- Click Solve or Evaluate Results
- Just like normal results
  - Only exception is no parameters right now 😞
Go to Town

• You can Duplicate a Branch to make more
  – RMB->Duplicate then change the worksheet for your next combination
  – No user defined yet

• Suggestions/comments
  – If you are comparing combinations, put some result object name:
    • Comb1: SX and Comb2: SX
  – Remember this for linear runs: unit loads
  – Works with transient as well
But Wait…

- They will not be adding capability to this feature in the future.
- The new way to do solution combinations will be through a DA system added to your project
  - Default configuration
- Right now it works pretty much the same.
- Fixes some things:
  - You can make a parameter out of a result min/max value!
  - User Defined Results can be used where they makes sense
- So, you can use the solution combination but we recommend you add a DA System
The API
Accessing the DA System

• A well documented API exists to allow you to make your own design assessment tools

• In the Project Toolbox->Analysis Systems there is a Design Assessment System
  – Drag and drop on your Static Structural or Transient Structural analysis

• Click on Setup Cell and look at properties
DA Setup

- You can do the default Solution Combination, the ASUS code checking, or User Defined.
- Solution Combination is just like what is built into ANSYS Mechanical.
  - As mentioned, we are now recommending that you use this because it is where things will get added in future releases.
- If you pick User Defined you have to point to your XML file and you will get your custom Design Assessment.
But...

- This is modern object oriented XML and Python
- You are hooking into the Workbench 1 infrastructure
- This is not simple scripting – no time to do a class today

However

- It is not as bad as it looks
  - The format is verbose, but truly the amount of work you need to do is pretty small
  - It is well documented
  - This is the future of customization, good way to get used to it

```xml
<?xml version="1.0" encoding="utf-8"?>
<!--Modified version of standard result combination -->
<DARoot ObjId ="1" Type="CAERep" Ver="2">
 <Attributes ObjId="2" Type="CAERepBase" Ver="2">
  <!--definition of attributes for re-use throughout the attribute groups.-->
  </Attributes>
 <AttributeGroups ObjId ="3" Type="CAERepBase" Ver="2">
  <!--grouping of attributes - used to define the available options in the attribute groups objects-->
  </AttributeGroups>
 <DAScripts ObjId="4" Type="DAScripts" Ver="2">
  <!--analysis script language & contents - used to define a script covering how the design assessment will be performed and a script used to obtain results-->  
  <Solve PropType="string">None</Solve>
  <Evaluate PropType="string">None</Evaluate>
  <DAData PropType="int">0</DAData>
  <CombResults PropType="int">1</CombResults>
  <SelectionExtra PropType="vector<string">""
  </SelectionExtra>
  </DAScripts>
 <Results ObjId="5" Type="CAERepBase" Ver="2">
  <!--definition of the available results and the available options in the results object.-->
  </Results>
 </DARoot>
```
Two Parts

- XML File
  - Defines things that WB1 needs to know

- Python Scripts
  - Grabs data from model
  - Does stuff with it
  - Brings data back into the model

- XML file is defined in help:
  - // Mechanical User's Guide // Features // Design Assessment :: 0 //
    The Design Assessment XML Definition File

- API for Scripts also in help:
  - // Mechanical User's Guide // Features // Design Assessment :: 0 //
    Design Assessment API Reference
XML FILES
XML File

- You define an XML file: Design Assessment XML Definition File
- Tells ANSYS Mechanical how to “fit” your DA into your model
- Three things are specified in file:
  - Attributes and Attribute Groups
  - Scripts to be run, if any
  - What results are produced and how to handle them

```xml
<?xml version="1.0" encoding="utf-8"?>
<Attributes ObjId="2" Type="CAERepBase" Ver="2">
</Attributes>

<AttributeGroups ObjId="3" Type="CAERepBase" Ver="2">
</AttributeGroups>

<DAScripts ObjId="4" Type="DAScripts" Ver="2">
</DAScripts>

<Results ObjId="5" Type="CAERepBase" Ver="2">
</Results>
```
Attributes and Attribute Groups

- Defines what goes into the Details view
- How you get information from the user to control what gets done
- Does not add object to tree automatically – Use Insert
- Does not give options of what attributes to add when inserting
  - User adds an Attribute Group objet
  - User Picks Group Type then Subtype to get to attributes
  - Headings in details are generic: Definition, Attribute 1, Attribute 2…
Attributes

• You create `<DAAttribute>` objects in the file for every item you want to prompt the user for

• `<DAAttribute ObjId="101" Type="DAAttribute" Ver="2">`
  – ObjID is the “handle” for the attribute – VERY IMPORTANT

• `<AttributeName PropType="string">Text</AttributeName>`
  – Specifies the name for the attribute
  – Does not change the headings in details 😞, is it’s own line in details

• `<AttributeType PropType="[Int|Double|Text|DropDown|Table|None]">Value</AttributeType>`
  – Specify the type of data that the user can enter
Attributes, Cont...

• `<Application PropType="string"> [Vertices|Lines|Surfaces|Solids|Geometry|All]</Application>`
  – Defines the type of geometry that the attribute applies to
  – If other than All, a scoping box is added to the details view.

• `<Validation PropType="vector&lt;string>“ [min,max|lenMax|T1,T2,T3…TN]</Validation>`
  – Specifies min,max for Int and Double
  – Max string length for String
  – Comma delimited list for dropdown values or table headings

• `<Default PropType="String">[Value|String|DropDownString] </Default>`
  – Specifies a default value for the attribute
  – leave blank for none

• Don’t panic: Help is very detailed on this!
Example Attributes

```xml
<Attributes ObjId="2" Type="CAERepBase" Ver="2">
  <DAAtribute ObjId="101" Type="DAAtribute" Ver="2">
    <AttributeName PropType="string">Stress Floor</AttributeName>
    <AttributeType PropType="string">Double</AttributeType>
    <Application PropType="string">Geometry</Application>
    <Validation PropType="vector<string">" 0,10000000
  </DAAtribute>

  <DAAtribute ObjId="102" Type="DAAtribute" Ver="2">
    <AttributeName PropType="string">Stress Ceiling</AttributeName>
    <AttributeType PropType="string">Double</AttributeType>
    <Application PropType="string">Geometry</Application>
    <Validation PropType="vector<string">" 0,10000000
  </DAAtribute>

  <DAAtribute ObjId="201" Type="DAAtribute" Ver="2">
    <AttributeName PropType="string">User Comments</AttributeName>
    <AttributeType PropType="string">Text</AttributeType>
    <Application PropType="string">All</Application>
  </DAAtribute>
</Attributes>
```
Attribute Group

- Users access attributes by choosing a group and a subgroup
  - Not done in the insert menu
- Specify:
  - Group ID:
    - `<DAAttributeGroup ObjId = “[ID]” Type="DAAttributeGroup” Ver="2”>`
  - Group name
    - `<GroupType PropType="string”>[NAME]”</GroupType>`
  - Subgroup name
    - `<GroupSubtype PropType="string”>[NAME]”</GroupSubtype>`
  - Attribute ID’s to put in the group
    - `<AttributeIDs PropType="vector&lt;unsigned int”>[ID1, ID2…Idn]”</AttributeIDs>`
- Key thing is to list the right ID’s for the `<AttributeIDs>`
- Also, pick descriptive names
  - Leave `<GroupSubtype>` blank or leave out if no subgroup
Example Attribute Group

<AttributeGroups ObjId ="3" Type="CAERepBase" Ver="2">
  <DAAttributeGroup ObjId="100001" Type="DAAttributeGroup" Ver="2">
    <GroupType PropType="string">Truncate Stresses</GroupType>
    <AttributeIDs PropType="vector&lt;unsigned int">101,102</AttributeIDs>
  </DAAttributeGroup>
  <DAAttributeGroup ObjId="100002" Type="DAAttributeGroup" Ver="2">
    <GroupType PropType="string">User Comments</GroupType>
    <GroupSubtype PropType="string">User Comments</GroupSubtype>
    <AttributeIDs PropType="vector&lt;unsigned int">201</AttributeIDs>
  </DAAttributeGroup>
</AttributeGroups>
**Script**

- Tells the program:
  - What Scripts to run for a Solve and for an Evaluate action
  - Options on allowing solution combination, results, or attributes to be inserted
  - Additional result columns to be put in solution selection worksheet
- `<Solve PropType="string">"[Script]"</Solve>`
  - Script name with full path
- `<Evaluate PropType="string">"[Script]" </Evaluate>`
  - Script name with full path
- `<DAData PropType="int"> [0|1] </DAData>`
  - Set to 1 to allow users to insert Attribute Groups or DA Results
- `<CombResults PropType="int"> [0|1] </CombResults>`
  - Set to 1 to allow the default Solution Combination as well as custom stuff
- `<CombExtra PropType="vector&lt;string">"[C1,C2…Cn] </CombExtra>`
  - Column names to show in Solution Selection Worksheet
  - User can type in info here if they would like
<Solve> vs. <Evaluate>

- **<Solve>**
  - Used to extract attribute and solution result data from the model
  - Runs any external post processing
  - Runs when user chooses “Solve”

- **<Evaluate>**
  - Extract results from post processor
  - Runs after <Solve> when “Solve” is chosen by user
  - Runs when user chooses “Evaluate All Results”

- **Why?**
  - User may run some sort of post processing that updates calculated values without needing to re-extract information from Mechanical
  - Just do an Evaluate
<DAScripts ObjId="4" Type="DAScripts" Ver="2">
  <Solve PropType="string">"c:\temp\truncstrs.py"</Solve>
  <Evaluate PropType="string">"c:\temp\intrunc.py"</Evaluate>
  <DAData PropType="int">1</DAData>
  <CombResults PropType="int">0</CombResults>
  <SelectionExtra PropType="vector&lt;string">" DeltaMin,DeltaMax
</SelectionExtra>
</DAScripts>
Results

• Tells the program what results are available to be added to the tree
• Works a lot like the input attributes, and uses them
• RMP->Insert->DA Result
  – Then you pick what result you want with type and subtype
• `<DAResult ObjId ="[ID]" Type="DAResult" Ver="2">`
  – One of these for each result group you define in you python script
• `<GroupType PropType="string">[GroupName]</GroupType>`
  – What to call it in the drop down
• `<GroupSubtype PropType="string"> [SubName] </GroupSubtype>`
  • What to call it in the drop down
  – `<AttributeIDs PropType="vector&lt;unsigned int>>">[ID1,ID2,..,Idn] </AttributeIDs>`
    • A list of the attributes to display
  – `<DisplayType PropType="string">ElmCont</AttributeIDs>`
    • The type of display, right now ElmCont is the only option
<Results ObjId="5" Type="CAERepBase" Ver="2">
  <DAResult ObjId ="100001" Type="DAResult" Ver="2">
    <GroupType PropType="string">Equiv Stress</GroupType>
    <GroupSubtype PropType="string">Truncated</GroupSubtype>
    <DisplayType PropType="string">ElmCont</DisplayType>
  </DAResult>
</Results>
Attributes vs. Result Attributes

- You define your attributes as DAAttributes and group them in two places!
  - AttributeGroups and DAResult
- User can add attributes in tree before solution
- Or they can add a DA Result
- Difference:
  - If user defines as Attribute Group
    - Values are set before solution
    - One Attribute value for all DA Results
    - Change attribute value will require a new solve of DA
  - If user defines as DA Result
    - Each DA Result can have its own Attribute Values
    - If you change the Attribute value, you only need to run Evaluate, not solve
Summary on XML

• Start simple and define a few key things
• Use RMB on Setup on system on project page to reread XML file as you build it
• In ANSYS Mecanical:
  – RMB->Insert->Attribute Group to add user prompts
  – RMB->Insert->DA Result on Solution to define what results to show
• Just the basics, use to learn
PYTHON SCRIPTS
Writing Scripts

• Not enough time to even sort of show this
• Not really hard, but each application is unique

• You should know some Python
  – http://docs.python.org/tutorial/
  – http://arstechnica.com/old/content/2008/12/getting-a-grip-on-python-six-ways-to-learn-online.ars

• Use API Reference in Help!!!!
  – // Mechanical User's Guide // Features // Design Assessment :: 0 // Design Assessment API Reference

• Don’t be afraid to ask your ANSYS Support provider for support
API: General Classes

• DesignAssessment
  – Parent class. General information about the DA objects

• Helper
  – Utilities that access ANSYS Mechanical specific information
  – Also accesses data that comes from XML files
  – Error handling, IO, logging, Units, etc…
  – Gets path to important files
  – Study this one, key things in there
API: Mesh Classes

- **GeometryMeshData**
  - Mesh information on nodes and elements

- **DAElement**
  - Access to a specific element: Type, Connectivity, section data, etc.

- **DANode**
  - Info about a specific node: location and elements it is attached to

- **SectionData**
  - Info on a specific elements section properties
API: Input Classes

• AttributeGroup
  – Used to access attributes added to the Mechanical tree

• Attribute
  – Used to get the specific values entered for an attribute

• Selection
  – Access what the user chose for the Solution Selection branch in the Mechanical tree

• Solution
  – Access each row of the SolutionSelection tree object
API: Solution Classes

- SolutionResult
  - Has a member that points to the result file for a given row in the Solution Selection table

- ValueStructure
  - Get single data values from model, mostly stuff in XML file
  - Set single data values in the Result objects

- ResultGroup
  - Collection of attributes for DA Result Objects
  - Like AttributeGroup, but for results

- ResultClass
  - Used to set a specific result value
Basic Steps

• Usual Steps in Solve Script: Getting Results from your Mechanical Run
  – Get any values that the user may have specified as DA attributes
  – Get results from your Mechanical run
    • Write an APDL macro using python that will:
      – Go into MAPDL
      – Extract info you want
      – Output in the format you want
    • Or have a pre-written macro and feed it parameters
      – Prefer creating macro so you have one less file to track
  • Write an *.inp file to resume and run your macro(s)
  • Execute MAPDL with input file
  – Now results are available for calculation in your evaluation script
Basic Steps

- Usual Steps in Evaluation Script: Using Results from your Mechanical Run to do Calculations
  - Read results from file created in Solve script
  - Get any attributes set in the DA Results added to the tree
  - Loop through DA Results, making calculations
    - Use MAPDL
      - Write macro then execute
    - Use Python
    - Use an external program
    - Any combination of the above
  - Create result objects
  - Load calculated values into Result objects for each type of result requested
  - Can be looped differently, whatever is best
Example: Stress Truncate

da_trunc.xml

da_trunc_solve.xml

da_trunc_eval.xml
Example: TsaiWu Values

- Uses user supplied criteria to do TsaiWu calculation in MAPDL and bring back results
Conclusion
R13: First Release

• The R13 release of DA was oriented towards supporting the needs of offshore users
  – But made open for other users
• Some limitations:
  – You need to be careful about paths in your script, no relative path yet
  – If you launch MAPDL in your scripts, you need two licenses
    • And set Options->tools->license preferences to disable sharing
  – Must run MAPDL to get calculated result information
  – Element results only (use ETABLE to get in MAPDL)
  – Windows only
• Most of these are planned to be addressed in R14
• Addition of many more features at R14
Lots of Power, Bit of a Learning Curve

- If you are still with us... You can see DA delivers:
  - Ability to gather User specified values
  - Allow user to specify what steps to work with
  - Execute scripts that grab user and result data and calculate new result values
  - Show results
- Best way to do your own custom post processing and stay in Mechanical
- Fits in the existing ANSYS Mechanical framework
- Very general nature takes getting used to
  - Groups and SubGroups
  - No predefined Attributes or results
- For many of us: the object oriented bit is hard to get our head around
  - Stick with it
  - There is a reason why it is so popular
Recommendations

- When working RMB on Setup cell and choose file, that rereads and redoes the DA
- Get an XML script editing tool
- Copy & paste existing guys
  - C:\Program Files\ANSYS Inc\v130\aisol\DesignSpace\DSPages\xml
  - We will post on The Focus: www.PADTINC.com/blog
- Lean on ANSYS support
- Share with the community!!!!!!
  - Post on XANSYS
  - Send to Sheldon for ANSYS.net
    - query1@ansys.net
Where to get more info?

• ANSYS, Inc. Ask the Experts Webinar:

• The help system:

• Look for detailed review of Truncate DA example in next weeks The Focus
  – [www.PADTINC.com/blog](http://www.PADTINC.com/blog)
Thank You…

- PADT Enjoys doing these webinars…
- Please consider us as your partner
- ANSYS Related
  - Training, Mentoring
  - Consulting Services
  - Customization
  - Sales (if in AZ, NM, CO, UT, NV)
- Stratasys 3D Printers and Systems
- CUBE HVPC Systems
- Product Development
  - High-end engineering with practical, real world application
- Rapid Prototyping
  - SLA, SLS, FDM, PolyJet, CNC, Soft Tooling, Injection Molding
- Help us by letting us Help you