

Some Thoughts on Usability of DOE Engineering Codes

Kyran Daniel Mish, Sandia/Albuquerque

Mod/Sim Forum, March 2, 2011

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.



Presentation Overview

- **The top three things to know about usability**
 - **The mental map**
 - **The developer's view**
 - **The user's view**
- **Some considerations of how engineers learn**
 - **Timoshenko's approach**
 - **Why engineers may avoid the math department**
- **Sierra usability examples**
 - **Sierra UI's past and present**
 - **Where we are going with Sierra UI next**

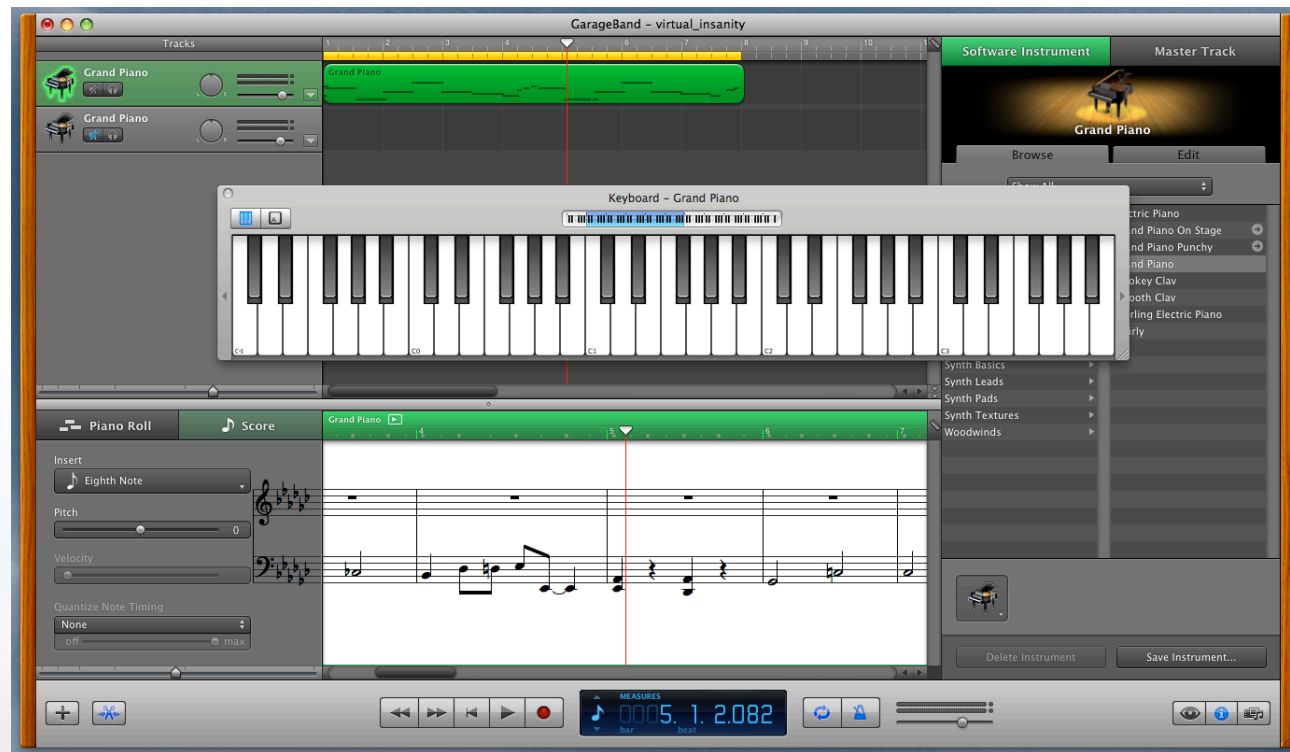


It's All About the User's Mental Map

- The primary goal of software usability is the creation in the user's mind of a *mental map* of the application's actual capabilities
 - A good UI creates an accurate map of what the program can do, and how it works
 - A bad UI (or no UI) fails to create this map, and forces the user to learn by rote practice what the app does
 - Both approaches can succeed or fail, but a good UI aids productivity, and that's always a good idea
- Let's look at some examples...

Good UI Example: Garage Band

- Given this UI, what does this application do?



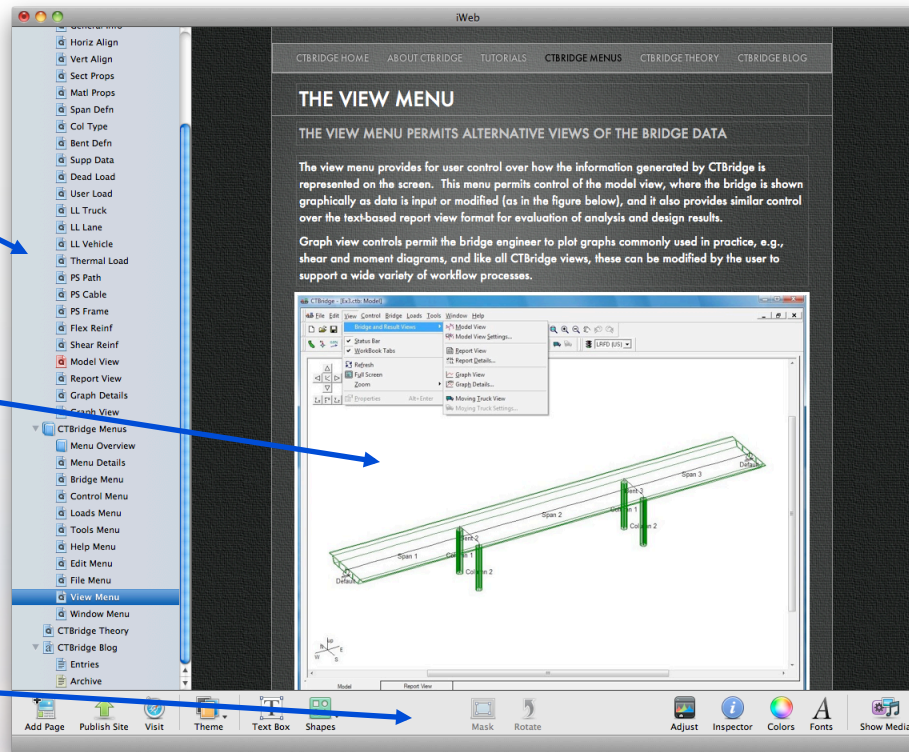
Another Good UI Example: iWeb

- Website development made easy via a good UI

Simple view of
website content
architecture

WYSIWYG
web content
development

Web content
management
tools readily
available here





The Developer's View

- **Developers know software capabilities firsthand**
 - **Developers possess privileged inside information about the application**
 - **Developers don't need a UI to form an accurate mental map of the software's capabilities**
 - **Developers know these capabilities directly, via access to the source code**
- **In essence, developers know too much**
 - **Developers need to be part of good UI design (e.g., feasibility aspects), but not the only part**



The User's View

- **Users learn software capabilities via the UI**
 - This knowledge of application capabilities may depend almost entirely on the quality of the UI
 - This is why a good UI is essential if an application is to gain broad and accurate use
 - The job of the UI designer is to aid the user in learning software capabilities through the UI
- **In essence, users know too little**
 - Thus good UI design generally starts with the users and works backward to the development process
 - Architecture-first process is one example



A Word About Engineering Education

- **Engineers learn via templates and cases**
 - **Learn series of concrete examples first (i.e., templates)**
 - **After this series, deduce more abstract principles**
 - **This approach is associated with Timoshenko**
 - **Contrast with abstraction-first math education view**
- **Case studies are important aspect of usability**
 - **These are often neglected in favor of UI technology**
 - **Best-practice is to include a coherent set of worked examples that serve as templates for effective use**
 - **Library of useful examples is an essential component of helping analysts learn how to use our software**

Textual UI Strengths and Weaknesses

- Two views of same simple problem

```
begin adagio procedure myProcedure
  begin time control
    begin time stepping block p0
      start time = 0.0
      begin parameters for adagio region myRegion
        time increment = 0.1
      end parameters for adagio region myRegion
    end time stepping block p0
    termination time = 1.15
  end time control

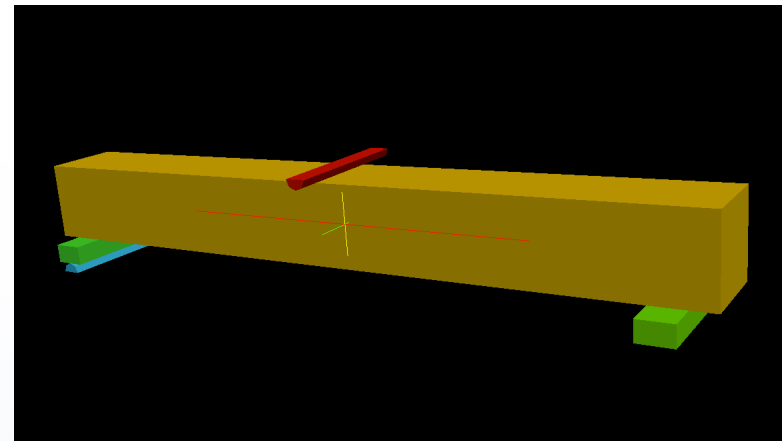
  begin adagio region myRegion

    use finite element model mesh

    begin contact definition contacts
      skin all blocks = on
      contact formulation type = dash
      begin interaction defaults
        friction model = sticky
        general contact = on
        self contact = off
        constraint formulation = node_face
      end
      BEGIN CONSTANT FRICTION MODEL sticky
      FRICTION COEFFICIENT = 0.3
    END
  end contact definition contacts

  begin user output
    node set = nodelist_2001 nodelist_2002
    compute global y_reaction as sum of nodal reaction(2)
  end user output

  begin user output
    node set = nodelist_2001
    compute global y_reaction_2001 as sum of nodal reaction(2)
  end user output
end
```



3-point bending test: not obvious from textual input, at least for the new user

Sierra Editor – Current Capability

Hyperlinks to Referenced Entities

Validation & Error Reporting

Automatic Syntax Completion

Contextual Help

The screenshot displays the Sierra Editor interface with the following components:

- Main Editor:** Contains Fortran code for a finite element model. A yellow tooltip is visible over the code, stating: "The interpretation of the geometry data stored in this database. Optional. Defaults to Cartesian."
- Outline View:** Located on the right, it shows a hierarchical tree of the model's entities, including materials, displacement fields, and specific test components like "3_point_bend_test1".
- 3D Model View:** At the bottom right, a 3D visualization of the 3-point bend test setup is shown, with a meshed beam supported by two rollers and a central load.
- Status Bar:** At the bottom, it shows the current line and column in the code: "[699 : 13] BEGIN PRESCRIBED DISPLACEMENT", along with the code name "raven" and version "4.18.3".

Outline View for Navigation

Visual Identification of Geometry & Boundary Conditions

Sierra Editor – Next Generation

Model Tree
Presentation
of the Input
Deck

Tree and Input
Deck Fully
Synchronized

Tree Based
Content
Creation &
Editing

Data Entry
Dialogs
Automatically
Generated from
Sierra XML

The screenshot displays the Sierra Editor interface with the following components:

- Outline (Model Tree):** A hierarchical tree view on the left showing the structure of the input deck, including finite element models, materials, procedures, and boundary conditions.
- Input Deck:** A central text editor showing the XML input deck for a 3-point bend test, including parameters like friction coefficient, capture tolerance, and fixed displacement definitions.
- Model View:** A 3D visualization of the finite element model, showing a rectangular block with a central notch and supports.
- XY Plot View:** A graph showing the displacement in the y-direction (displacement_y) versus position, with a red line representing the data.
- Data Entry Dialog:** A dialog box for editing a fixed displacement boundary condition, showing fields for block, surface, components (x, y, z), and node set (nodelist_1000).

```
friction coefficient = 0
capture tolerance = 2e-5
tension release = 5 iterations = 4
normal tolerance = 4e-5
end interaction int5

end contact definition contacts
# - Block id 2 had name 2
begin fixed displacement
block = block_1
components = x y z
end fixed displacement
# - NodeSet id 1000 had name ns1000
begin fixed displacement
node set = nodelist_1000
components = X Z
end fixed displacement
# - NodeSet id 3 had name 3
begin fixed displacement
node set = nodelist_3
components = X Y Z
end fixed displacement
```

Fully Integrated Sierra CAE Environment

Fully Visual
Based Model
Building

Integrated Job
Submission to
Remote HPC
Platforms

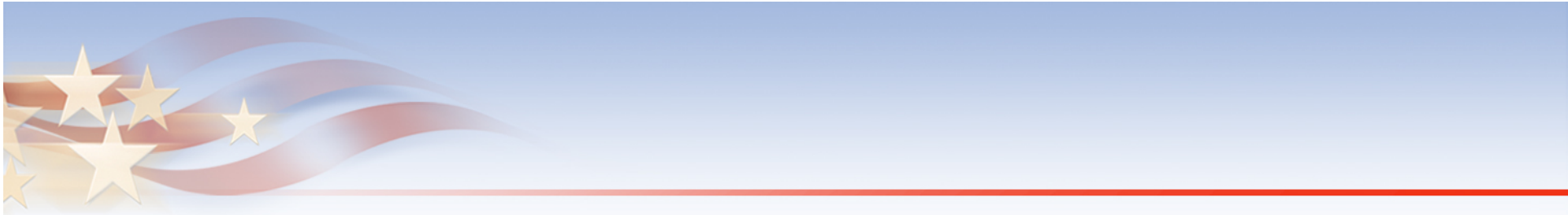
Power Users
Still Have
Access to
Input Decks

The screenshot displays the SierraEditor interface. On the left, the 'Outline' tree shows a hierarchical structure of the model, including 'finite element models', 'functions', 'geometry', 'materials', 'procedures', and 'constraints'. The main 'Model View' shows a 3D visualization of a pink rectangular block with a central notch, supported by two blue cylindrical rollers. A coordinate system (X, Y, Z) is visible in the top left of the model view. An 'Edit Configuration and Launch' dialog box is open in the foreground, allowing users to configure job submission parameters. The dialog includes fields for 'Name', 'Code', 'Machine', 'Queue', 'Remote Directory', 'Number Of Processors', 'Job Time', and 'Account'. The 'Machine' is set to 'glory - Sandia Machines' and the 'Queue' is set to 'adagio-glory - Sandia Machines'. The 'Number Of Processors' is set to 256. The 'Job Time' is set to 12 hours. The 'Account' is set to 'FY115555'. The dialog also has 'Apply', 'Revert', 'Close', and 'Run' buttons.



The Sierra UI Next Steps

- **Harden prototype functions for broad use**
 - **All functions shown work now**
 - **Some interesting new functions TBA soon!**
- **Develop standard library of sample problems**
 - **Develop and deploy DoD-centric templates**
 - **Web presence is one current sticking point**
- **Rapid progress since Fall TCG meeting**
- **Need to develop appropriate UI for UQ**



Questions and Comments

