Introduction to OpenSees and Tcl

Frank McKenna

NEES Hybrid Simulation Workshop
Berkeley, CA

What is OpenSees?

The Open System for Earthquake Engineering Simulation is:
• A software framework for developing sequential, parallel and grid-enabled finite element applications in earthquake engineering.
What is a Software Framework?

- A framework is NOT an executable.
- A framework is a set of cooperating software components for building applications in a specific domain.
- A framework dictates the architecture of the application.
- A framework is based on the assumption that a good design will cover the majority of applications within the domain.
- Loose-coupling of components within the framework is essential for extensibility and re-useability for applications.

Conceptual Approach for Simulation

- Computation
  - Algorithms, Solvers, Parallel/distributed computing
- Information Technology
  - Software framework, Databases, Visualization, Internet/grid computation
- Models
  - Simulation models, Performance models, Limit state models
  - Material, component, system models
Main Abstractions in OpenSees

ModelBuilder → Domain → Analysis

- ModelBuilder: Constructs the objects in the model and adds them to the domain.
- Domain: Monitors user defined parameters in the model during the analysis.
- Analysis: Moves the model from state at time t to state at time t + dt.
- Recorder: Holds the state of the model at time t and (t + dt).

What Types of Recorder

- **Recorder**
  - Element
  - Node
  - EnvelopeNode
  - EnvelopElement

- **DataOutputHandler**
  - StreamHandler
  - FileHandler
  - DatabaseHandler

- **Renderer**
  - X11
  - OpenGL
  - VRML
  - XML

- **Database**
  - File
  - MySQL
  - Oracle
  - NEES
What is in a Domain

Some Other Classes associated with Elements:
Other Classes associated with Elements:

Material

Uniaxial
- Elastic
- ElasticPP
- Hardening
- Concrete
- Steel
- Hysteretic
- PY-TZ-QZ
- Parallel
- Series
- Gap
- Fatigue

nD
- Elastic
- J2
- TemplateElasto-Plasto
- FluidSolidPorous
- PressureMultiYield(dependent, independent)

Section
- Elastic
- Fiber

Analysis

What is an Analysis

Analysis
- StaticAnalysis
- TransientAnalysis

CTest

CHandler
- Penalty
- Lagrange
- Transformation

Numberer
- RCM
- MinDegree

AnalysisModel
- EquiSolnAlgo
- Linear
- NewtonRaphson
- ModifiedNewton
- Broyden
- BFGS
- KrylovNewton
- NewtonLineSearch

SolnAlgorithm
- StaticIntegrator
- LoadControl
- DispControl
- ArcLength
- MinUnbalDispNorm
- TransientIntegrator
- Newmark
- HHT
- CentralDifference
- AlphaOS, NewmarkHybrid...

Integrator

SystemOfEqn
- StaticIntegrator
- BandGeneral
- BandSPD
- ProfileSPD
- SparseGeneral
- SparseSymmetric
Classes for Parallel & Distributed Processing

- **Channel** objects for communicating between processes
- **ObjectBroker** for creating blank objects upon which `recvSelf()` called
- **Shadow** (Proxy) objects to hide parallelism from existing objects
- **Actor** objects to sit on a remote process & process task requested
- **Machine** objects to start/manage processes (returns Channel to Shadow objects)

### OpenSees.exe

- OpenSees is an Open-Source Software Framework for developing Nonlinear Finite Element Applications for both sequential and parallel environments.

- OpenSees.exe is an extension of the Tcl interpreter for finite element analysis which uses this framework. It is an example of an application that can be developed using the framework.
What is Tcl

- Tcl is a string based scripting language.
- Variables and variable substitution
- Expression evaluation
- Basic control structures (if, while, for, foreach)
- Procedures
- File manipulation
- Sourcing other files.

Tcl

- Tcl scripts are made up of commands separated by newlines or ;
- Command syntax:
  
  command arg1 arg2 ...

- Help
  2. Practical Programming in Tcl and Tk, Brent B. Welch, Prentice Hall.
Example Tcl:

```tcl
>set a 1
>1
>set b a
>a
>set b $a
>1
>expr 2 + 3
>5
>expr 2 + $a
>3
>set b [expr 2 + $a]
>3

for {set i 1} {i < 10} {incr i 1} {
    puts “i equals $i”
}
set sum 0
foreach value {1 2 3 4} {
    set sum [expr $sum + $value]
}
set $sum
>10
>proc guess {value} {
    global sum
    if {$value < $sum} {
        puts “too low”
    } else {
        if {$value > $sum} {
            puts “too high”
        } else {
            puts “you got it!”
        }
    }
}
> guess 9
  too low
>
>proc sum {a b} {
    return [expr $a + $b]
}
>sum 2 3
>5
>set c [sum 2 3]
>5

>set fileId [open tmp w]
>??
>puts $fileId “hello”
>close $fileId
>type tmp
hello
>
>source Example1.tcl
```

Commands to Tcl for OpenSees

- For OpenSees we have added commands to Tcl for finite element analysis:
  1. Modeling – create nodes, elements, loads and constraints
  2. Analysis – specify the analysis procedure.
  3. Output specification – specify what it is you want to monitor during the analysis.
Model Generation:
*Adds the modelling commands to the interpreter.

**BasicBuilder**

```
model Basic -ndm ndm? <-ndf ndf?>
```

This command now adds the following commands to the interpreter:

- node
- element
- pattern
- fix
- equalDOF
- pattern
- fix
- fixX
- fixY
- fixZ
- uniaxialMaterial
- load
- eleLoad
- section
- sp
- mass
- block2D
- block3D
- patch
- layer
- fiber
- nDMaterial
- geomTransf

### Analysis

- **Solver**
  - StaticAnalysis
  - TransientAnalysis

- **SystemOfEqn**
  - BandGeneral
  - BandSPD
  - ProfileSPD
  - SparseGeneral
  - SparseSymmetric

- **Integrator**
  - StaticIntegrator
  - LoadControl
  - DispControl
  - ArcLength
  - MinUnbalDispNorm
  - Newmark
  - HHT

- **SolnAlgorithm**
  - EquiSolnAlgo
    - Linear
    - NewtonRaphson
    - ModifiedNewton
    - Broyden
    - BFGS
    - KrylovNewton

- **AnalysisModel**
  - Penalty
  - Lagrange
  - MinDegree
  - Transformation

- **Numberer**
  - Penalty
  - Lagrange
  - MinDegree
  - Transformation

- **CHandler**
  - Penalty
  - Lagrange
  - MinDegree
  - Transformation

- **analyze**

- **constraints**
- **numberer**
- **algorithm**
- **integrator**
- **system**
- **analysis**
Example Model:

```
model Basic -ndm -ndf 2
node 1 0.0 0.0
node 2 144.0 0.0
node 3 168.0 0.0
node 4 72.0 96.0
fix 1 1 1
fix 2 1 1
fix 3 1 1
uniaxialMaterial Elastic 1 3000.0
element truss 1 1 4 10.0 1
element truss 2 2 4 5.0 1
element truss 3 3 4 5.0 1
Pattern Plain 1 “Linear” {
  load 4 100.0 -50.0
}
```

Example Analysis:

- **Static Nonlinear Analysis with LoadControl**
  - constraints transformation
  - numberer RCM
  - system BandGeneral
  - test NormDispIncr 1.0e-6 6 2
  - algorithm Newton
  - integrator LoadControl 0.1
  - analysis Static
  - analyze 10

- **Transient Nonlinear Analysis with Newmark**
  - constraints transformation
  - numberer RCM
  - system BandGeneral
  - test NormDispIncr 1.0e-6 6 2
  - algorithm Newton
  - integrator Newmark 0.5 0.25
  - analysis Transient
  - analyze 2000 0.01
And Why do Finite Element Analysis
NCEER frame tested at the Taiwan facility

OpenSees Community Website

- Web site: [http://opensees.berkeley.edu/](http://opensees.berkeley.edu/)
- User Pages
  - Command Manual
  - Examples Manual
  - Browse the Source Code
  - Message Board
  - Bug Reporting!