Introduction to OpenSees and Tcl

Frank McKenna

NEES Hybrid Simulation Workshop
Berkeley, CA
December 12-13, 2005
What is OpenSees?

The **Open System for Earthquake Engineering Simulation** is:

- A software *framework* for developing sequential, parallel and grid-enabled simulation applications in earthquake engineering using finite element methods.
- A communication mechanism within PEER and NEES, and beyond, for exchanging and building upon research accomplishments.
- As open-source software, it has the potential for being a community code for earthquake engineering.

http://opensees.berkeley.edu/phpBB2/index.cgi
What is a Software Framework?

- A Framework is not a ‘code’, it is not an executable.
- A Software framework is a set of cooperating software components (classes) that work for building applications in specific domains. Programmers use inheritance and delegation to extend the framework.
- Loose coupling between components is essential for Extensibility and Reusability of the framework.
- Components for Finite Element Simulation
Main Abstractions in OpenSees

ModelBuilder

Domain

Holds the state of the model at time $t$ and $(t + dt)$

Recorder

Monitors user defined parameters in the model during the analysis

Analysis

Moves the model from state at time $t$ to state at time $t + dt$
What Types of Recorder

Recorder

Element
Node
EnvelopeNode
EnvelopeElement
Database
FilePlotter
Display

DataOutputHandler

StreamHandler
FileHandler
DatabaseHandler

Renderer

X11
OpenGL
VRML

Database

File
MySQL
Oracle
NEES
What is in a Domain

Domain

Element  Node  MP_Constraint  SP_Constraint  LoadPattern  TimeSeries

ElementalLoad  NodalLoad  SP_Constraint

Truss  ZeroLength  ElasticBeamColumn  NonlinearBeamColumn(force, displacement)  BeamWithHinges  Quad(std, bbar, enhanced, u-p)  Shell  Brick(std, bbar, 20node, u-p, u-p-U)  Joint  ExperimentalElement

Constant  Linear  Rectangular  Sine  Path
Some Other Classes associated with Elements:

- GeomTransformation
  - Linear
  - Pdelta
  - Corotational

Element in Global System

Geometric Transformation

Element in Basic System
Other Classes associated with Elements:

- Material
  - Uniaxial
    - Elastic
    - ElasticPP
    - Hardening
    - Concrete
    - Steel
    - Hysteretic
    - PY-TZ-QZ
    - Parallel
    - Series
    - Gap
  - nD
    - Elastic
    - J2
    - TemplateElasto-Plasto
    - FluidSolidPorous
  - Section
    - Elastic
    - Fiber
    - PressureMultiYield (dependent, independent)
What is an Analysis
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)
Domain Classes for Parallel FE

- **MovableObject**
  - sendSelf(Channel, ..)
  - recvSelf(Channel, ..)

- **Domain**

- **TimeSeries**
- **LoadPattern**
- **SP_Constraint**
- **MP_Constraint**
- **Element**

- **ElementalLoad**
- **NodalLoad**
- **SP_Constraint**

- **PartitionedDomain**

- **Subdomain**
  - NonlinearBeamColumn
  - BeamWithHinges
  - Quad (std, bbar,)
  - Brick (std, bbar)
  - Shell

- **GraphPartitioner**
- **DomainPartitioner**

- **LoadBalancer**

- **Partitioner**
  - **Metis**
  - **ParMetis**
Analysis Classes for Parallel FE

MovableObject
- sendSelf(Channel, ..)
- recvSelf(Channel, ..)

Analysis
- StaticAnalysis
- TransientAnalysis

DomainDecompAnalysis
- StaticDDAnalysis
- TransientDDAnalysis

SubstructuringAnalysis

SolnAlgorithm
- EquiSolnAlgo
  - Linear
  - NewtonRaphson
  - ModifiedNewton
  - Broyden
  - BFGS
  - KrylovNewton
- StaticIntegrator
  - LoadControl
  - DispControl
  - ArcLength
  - MinUnbalDispNorm
- TransientIntegrator
  - Newmark
  - HHT

Integrator

SystemOfEqn
- BandGeneral
- BandSPD
- ProfileSPD
- SparseGeneral
- SparseSymmetric
- DistributedSparse

Solver
- Lapack(Gen, Band, ..)
- ProfileSPD
- SuperLU
- Umfpack
- SparseSym
- DistributedSuperLU
- StaticCDProfileSPD
- FETI_ProfileSPD

Numberer
- RCM
- ParallelNumberer

ErrorHandler
NEESit Simulation

Simulation Portal

OpenSees NCTP Plugin

Compute Resources

Data Repository
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.
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
  1. [http://dev.scriptics.com/scripting/primer.html](http://dev.scriptics.com/scripting/primer.html)
  2. Practical Programming in Tcl and Tk, Brent B. Welch, Prentice Hall.
Example Tcl:

```tcl
>proc sum {a b} {
    return [expr $a + $b]
}
>set a 1
>1
>set b $a
>a
>set b $a
>5
>set c [expr 2 + $a]
>5
>for {set i 1} {$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
>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.

• For help
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
- mass
- block2D
- element
- fix
- block3D
- pattern
- fixX
- patch
- fix
- fixY
- layer
- equalDOF
- fixZ
- fiber
- pattern
- uniaxialMaterial
- load
- nDMaterial
- eleLoad
- section
- sp
- geomTransf
**OpenSees**

```
constraints type? args...
numberer type? args...
algorithim type? args...
integrator type? args...
system type? args...
analysis type? args...
analyze args ...```
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

Centerline model and model with joint comparison
Getting OpenSees

• Web site:  http://opensees.berkeley.edu/

• User Pages
  Download Center
  Documentation
  Browse the Source Code
  Message Board
  Bug Reporting!