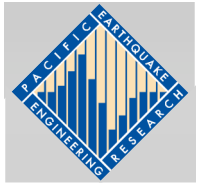
OpenSees

Introduction to OpenSees and Tcl/Tk

Frank McKenna
UC Berkeley
OpenSees Days 2012

http://opensees.berkeley.edu/wiki/index.php/OpenSeesDays2012



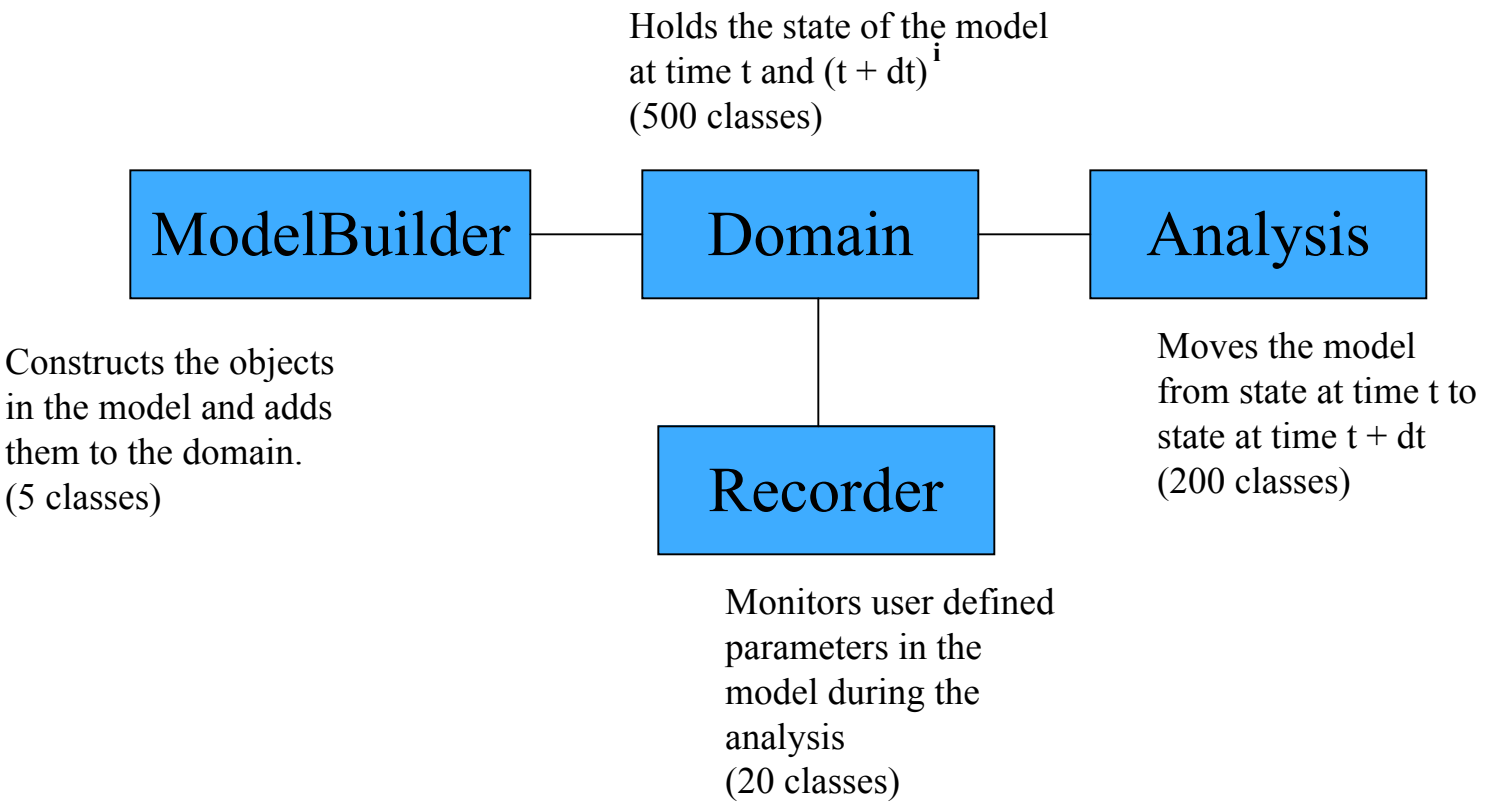
Outline of Presentation

- Overview of OpenSees the **FRAMEWORK**
- Introduction to Tcl Programming Language and Tcl interpreters
- Introduction to OpenSees.exe the **APPLICATION**

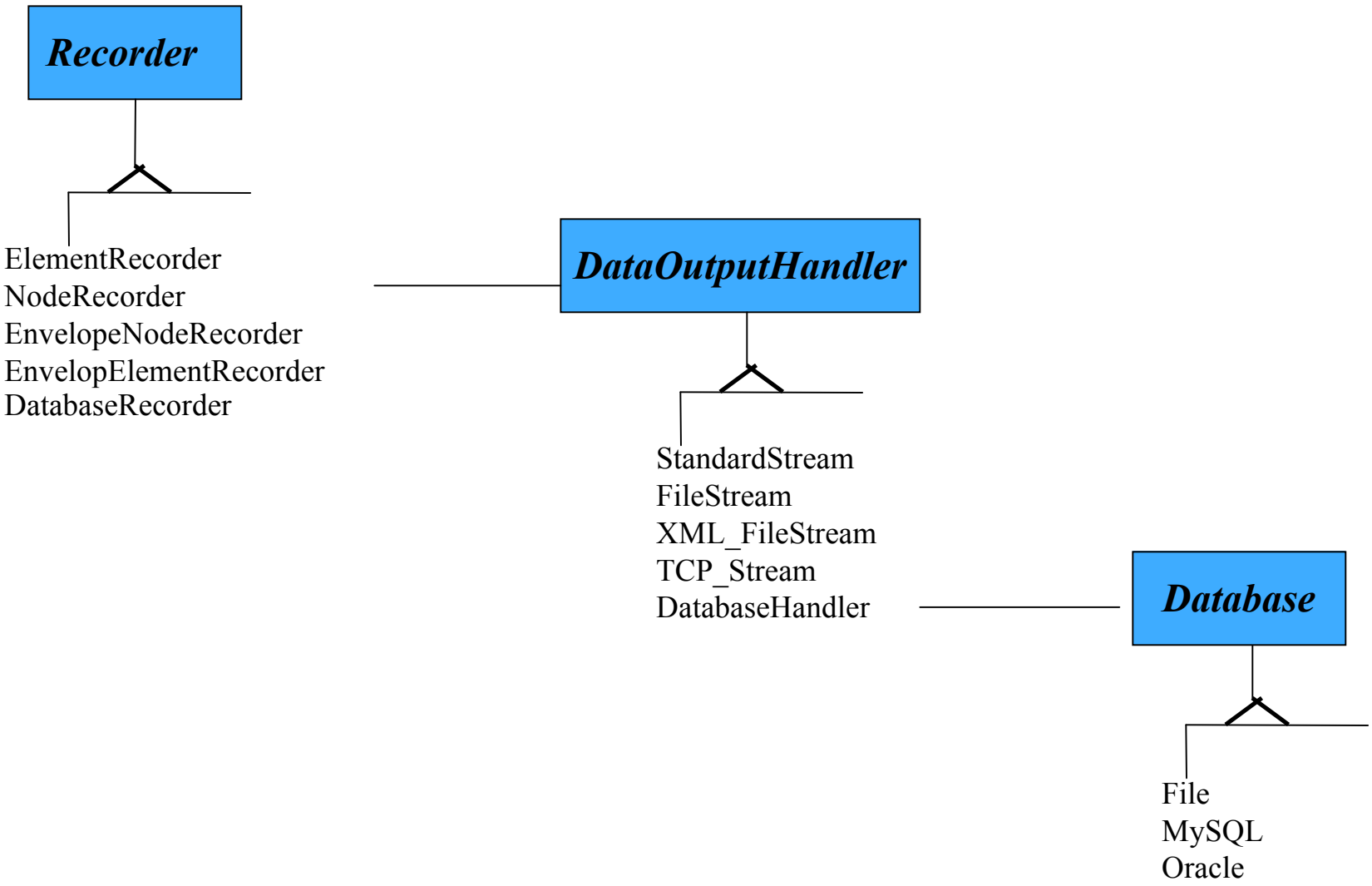
OpenSees is a Software Framework

- A framework is **NOT an executable**.
- A ***framework*** IS a set of cooperating software components for building applications in a specific domain.
- The OpenSees framework is written primarily in the object-oriented language C++; though other languages namely C and Fortran are also used.
- The abstract classes in the OpenSees framework define the interface. The concrete subclasses that exist in the framework provide the implementations.
- Other classes can be provided to extend the capabilities of the framework by developers using DLL's or providing the source code to the OpenSees repository.
- Currently over 1000 classes in the OpenSees framework.

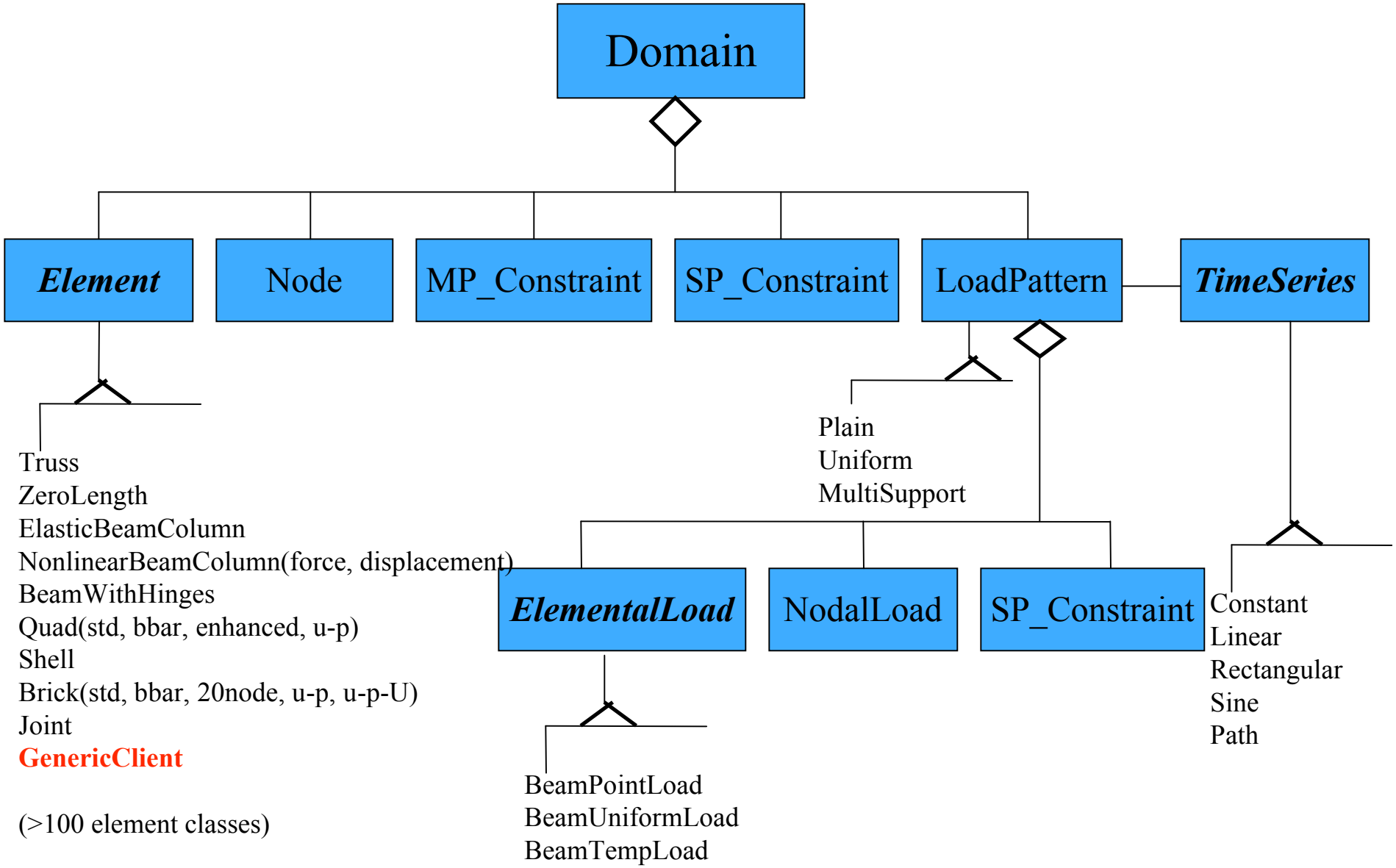
Main Abstractions in OpenSees Framework



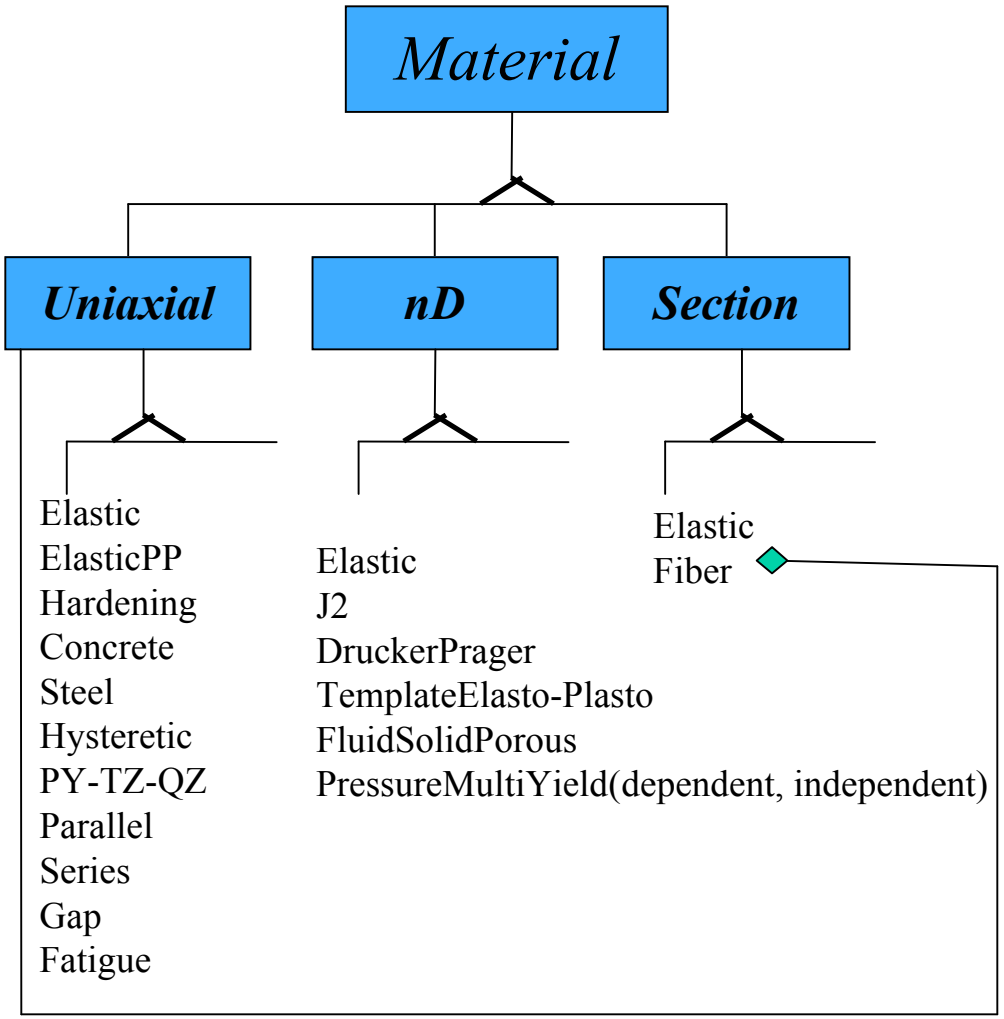
Recorder Options



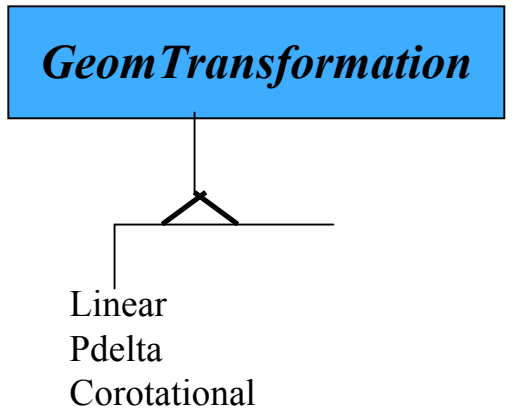
What is in a Domain?



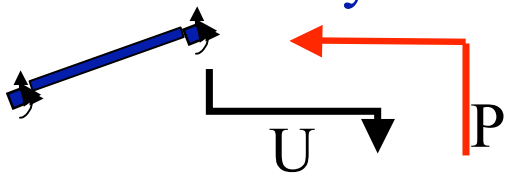
Some Other Classes associated with Elements:



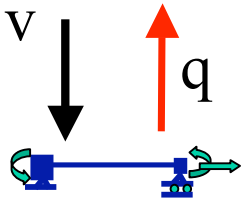
(over 250 material classes)



Element in Global System

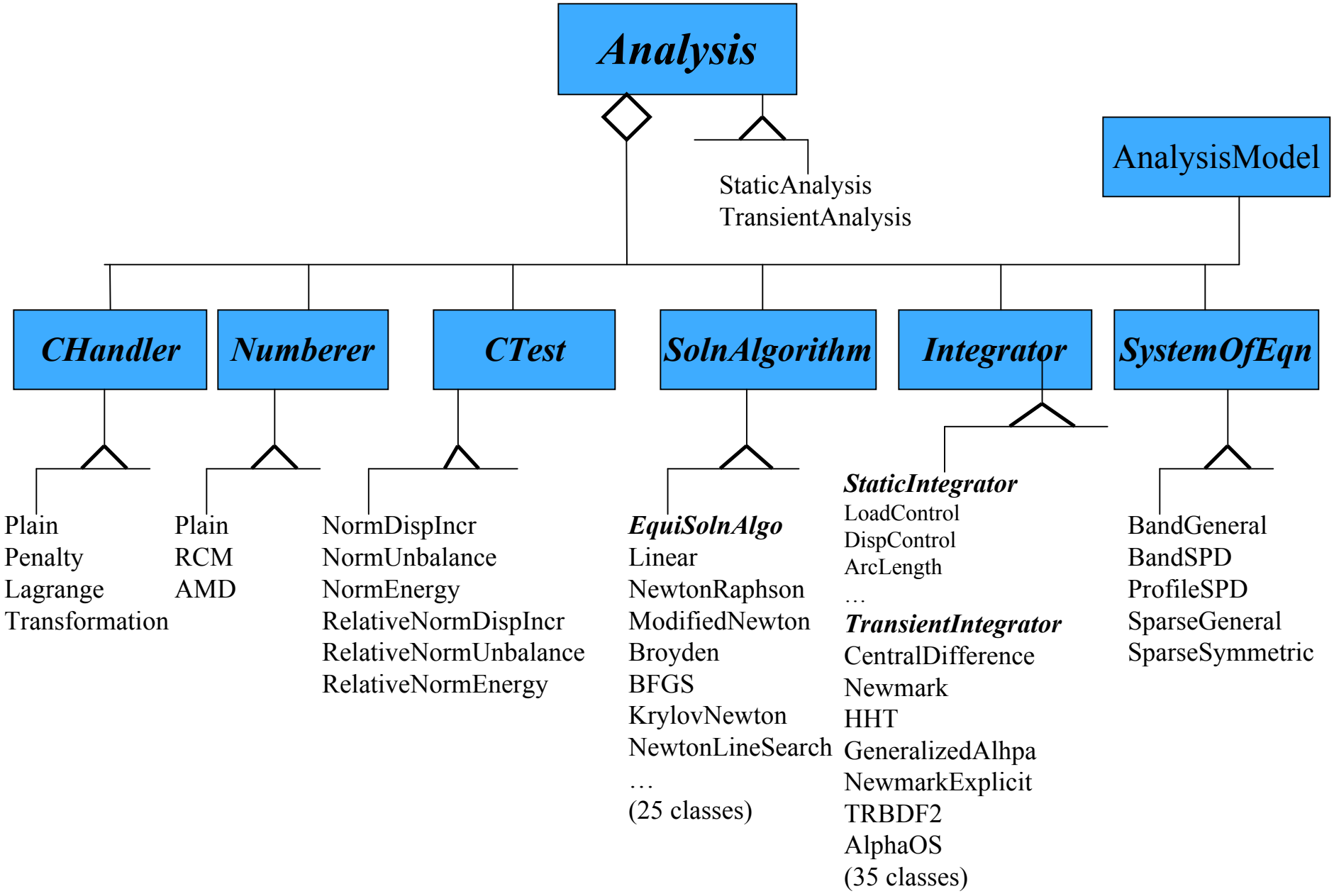


Geometric Transformation



Element in Basic System

What is an Analysis?



How Do People Use the OpenSees Framework?

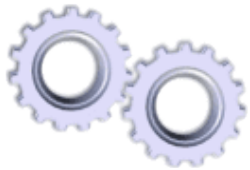
- Provide their own main() function in C++ and link to framework.

- Use OpenSees interpreter**S**. These are extensions of the Tcl interpreters, tclsh and wish, for performing finite element analysis.
    1. OpenSees.exe
    2. OpenSeesTk.exe
    3. OpseseesSP.exe
    4. OpenSeesMP.exe

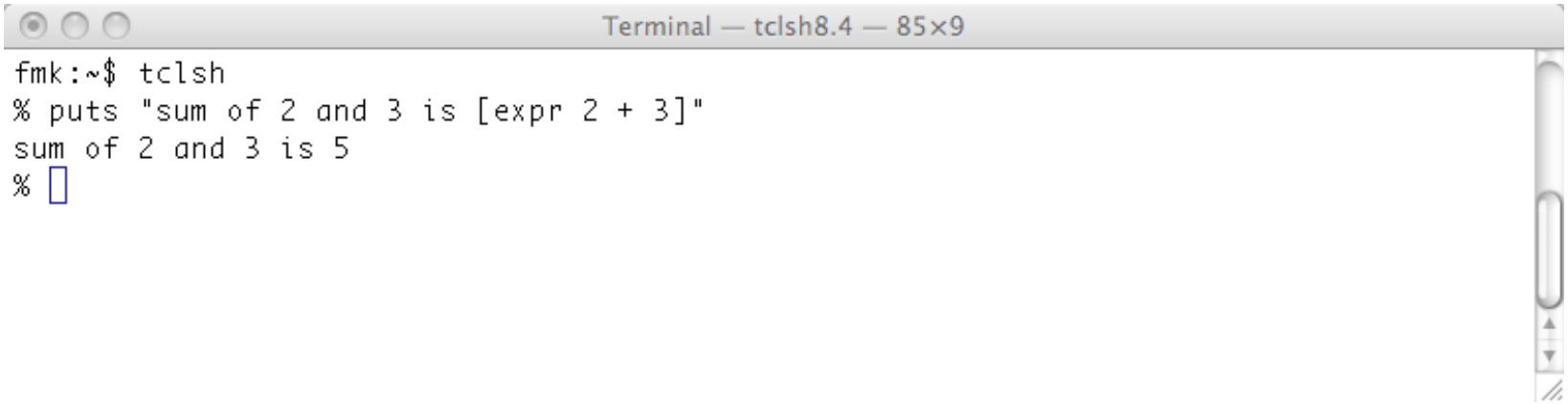
Tcl Interpreters

- **wish** and **tclsh** **are tcl interpreters**.
  - Interpreters (Perl, Matlab, Ruby) are programs that execute programs written in a programming language immediately.
  - There is no separate compilation & linking.
  - An interpreted program runs slower than a compiled one.

puts “sum of 2 and 3 is [expr 2 + 3]”



sum of 2 and 3 is 5



What is Tcl

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

- Comand syntax:

  command arg1 arg2 …

- Help

1. http://www.tcl.tk/man/tcl8.5/tutorial/tcltutorial.html

Example Tcl

•*variables & variable substitution*

```
>set a 1
1
>set b a
a
>set b $a
1
```

•*expression evaluation*

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

•*lists*

```
>set a {1 2 three}
1 2 three
>set la [llength $a]
3
>set start [lindex $a 0]
1
>lappend a four
1 2 three four
```

•*file manipulation*

```
>set fileId [open tmp w]
??
>puts $fileId "hello"
>close $fileID
>type tmp
hello
```

•***sourcing other files***

```
>source Example1.tcl
```

•*procedures & control structures*

```
> 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]
}
>puts $sum
10
>proc guess {value} {
    global sum
    if {$value < $sum} {
       puts "too low"
    } else {
       if {$value > $sum} {
          puts "too high"
       } else { puts "you got it!"}
   }
}
 > guess 9
too low
```

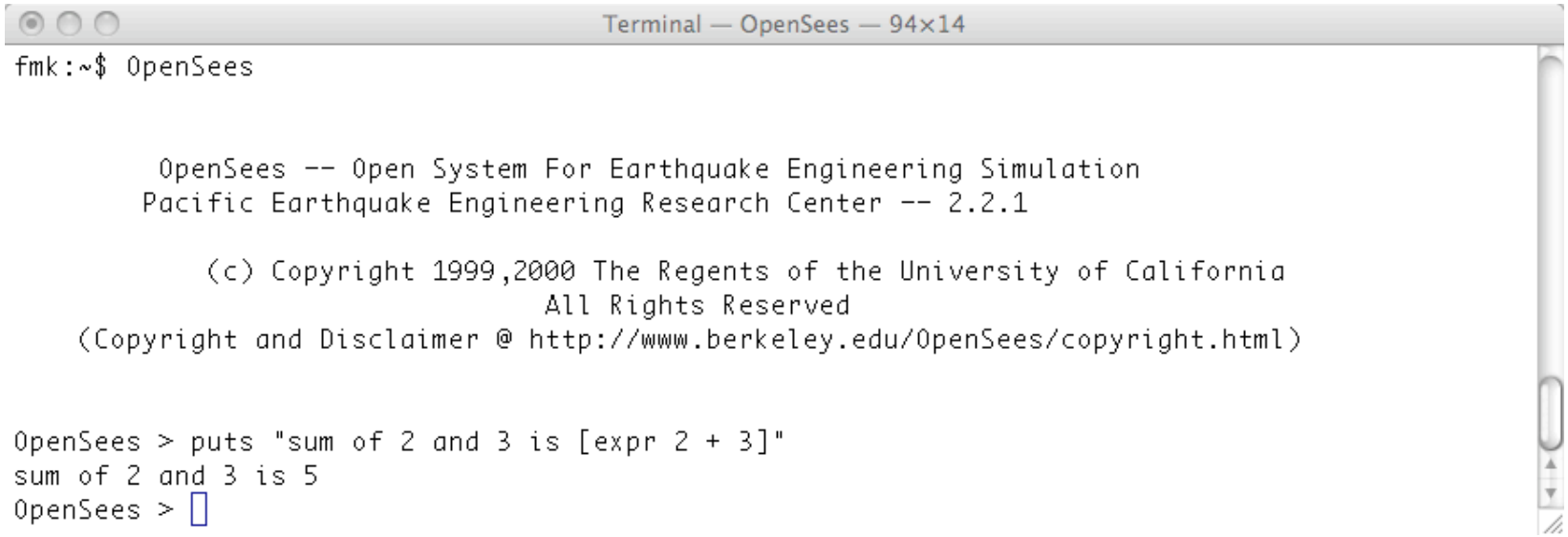
OpenSees Interpreters

- The OpenSees interpreters are tcl interpreters which have been extended to include commands 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.

- Being interpreters, this means that the files you create and submit to the OpenSees interpreters are not input files. You are creating and submitting PROGRAMS.

OpenSees.exe

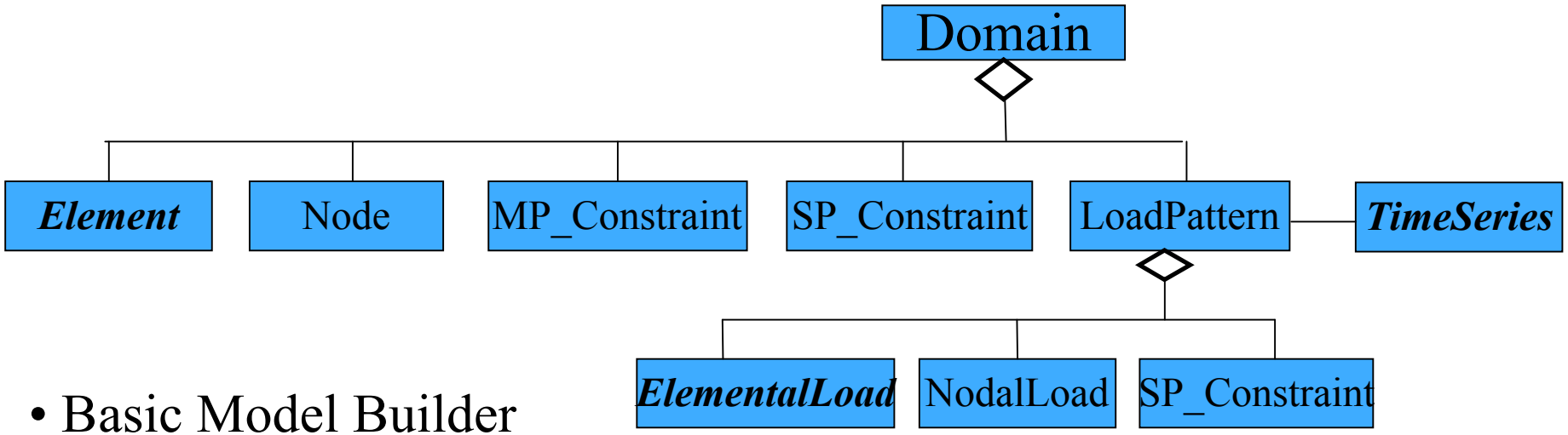
- An interpreter that extends tclsh for FE analysis.



WARNING: There is no GUI!

model Command

*Adds the modeling commands to the interpreter.



- Basic Model Builder

model Basic –ndm ndm? <-ndf ndf?>

This command now adds the following commands to the interpreter:

node mass element equalDOF fix fixX fixY fixZ

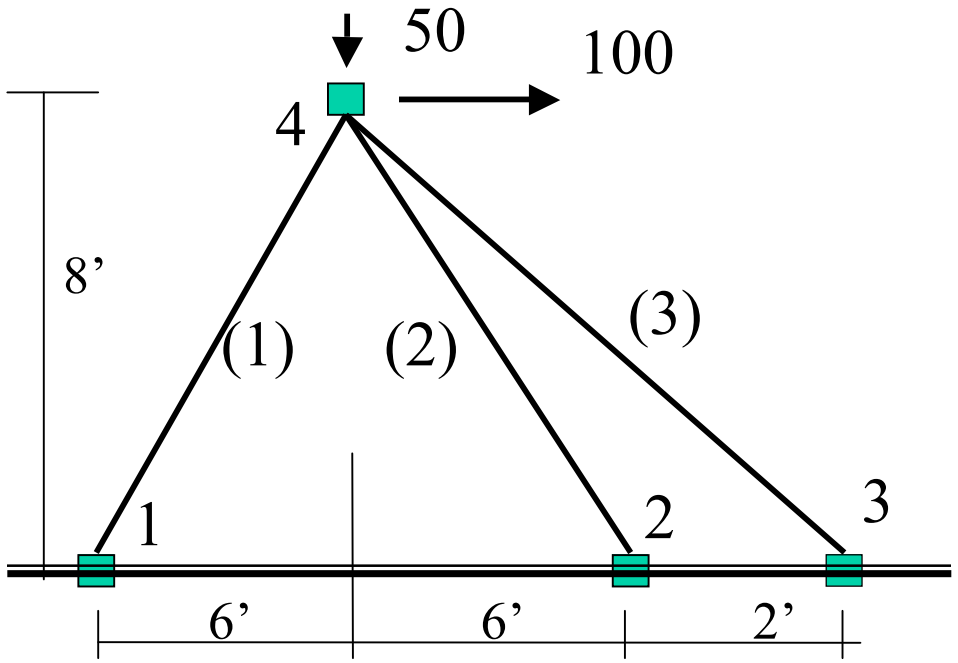
pattern timeSeries load eleLoad sp

uniaxialMaterial nDMaterial section geomTransf

fiber layer patch block2D block3D

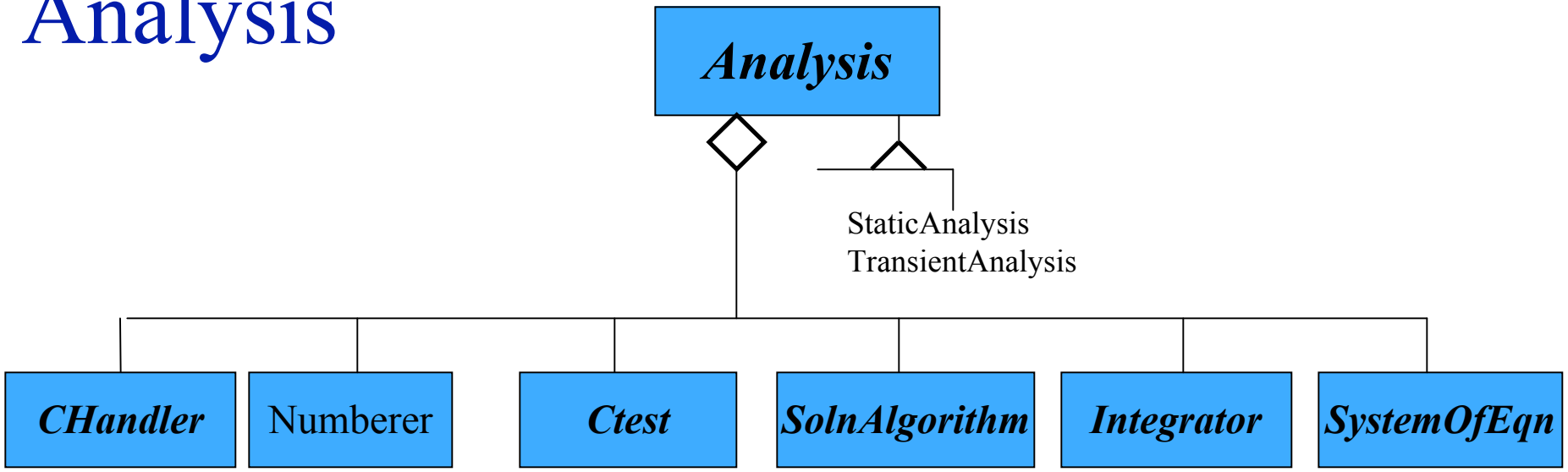
Truss example:

```
model Basic -ndm 2 -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
timeSeries Linear 1
pattern Plain 1 1 {
   load 4 100.0 -50.0
}
```



	E	A
1	3000	10
2	3000	5
3	3000	5

Analysis



handler type? args…

numberer type? args…

test type? args…

algorithm type? args…

integrator type? args…

system type? args…

analysis type? args..

analyze args …

Example Analysis:

- Static Nonlinear Analysis with LoadControl

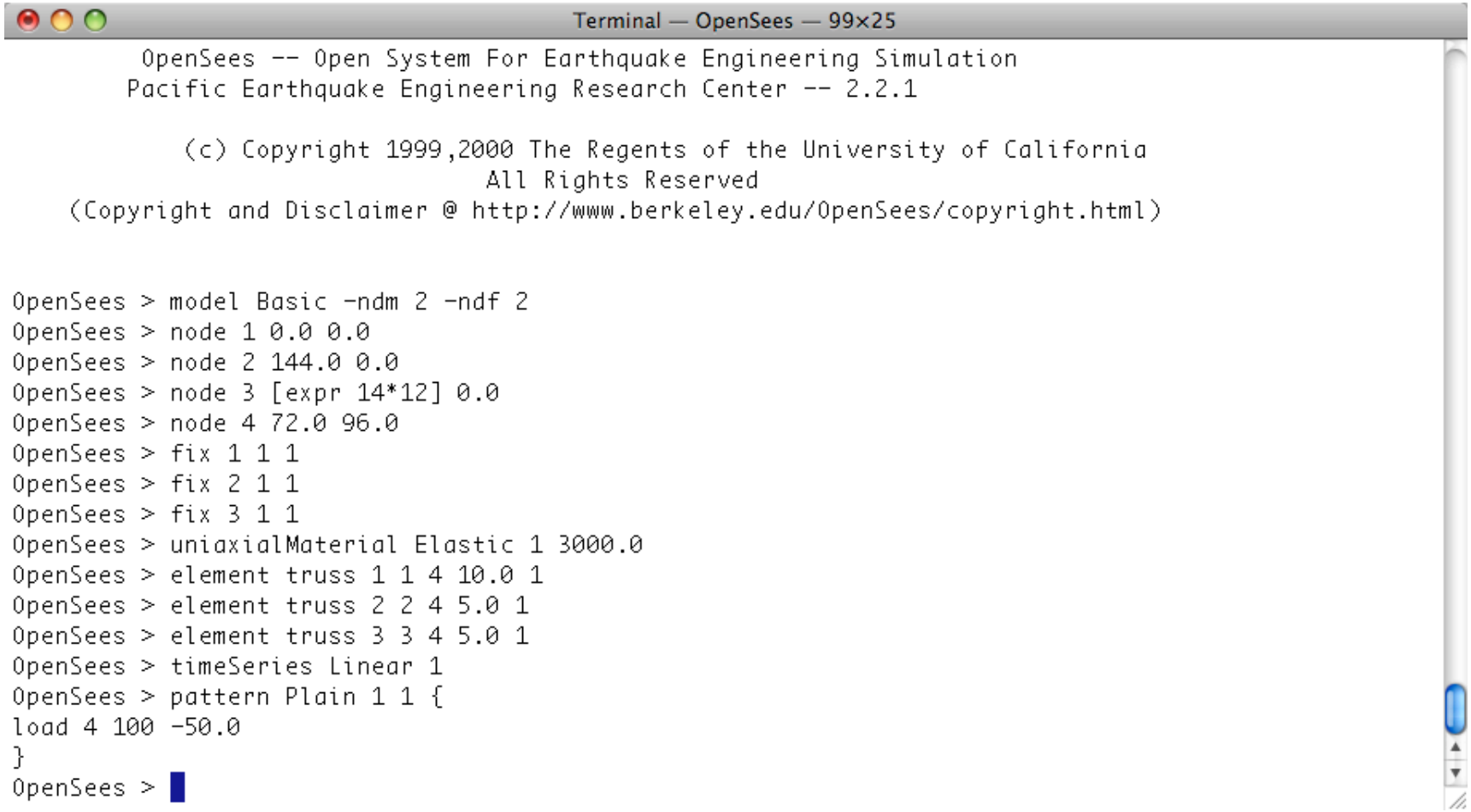
```
constraints Transformation
numberer RCM
system BandGeneral
test NormDispIncr 1.0e-6 6 2
algorithm Newton
integrator LoadControl 0.1
analysis Static
analyze 10
```

- 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
```

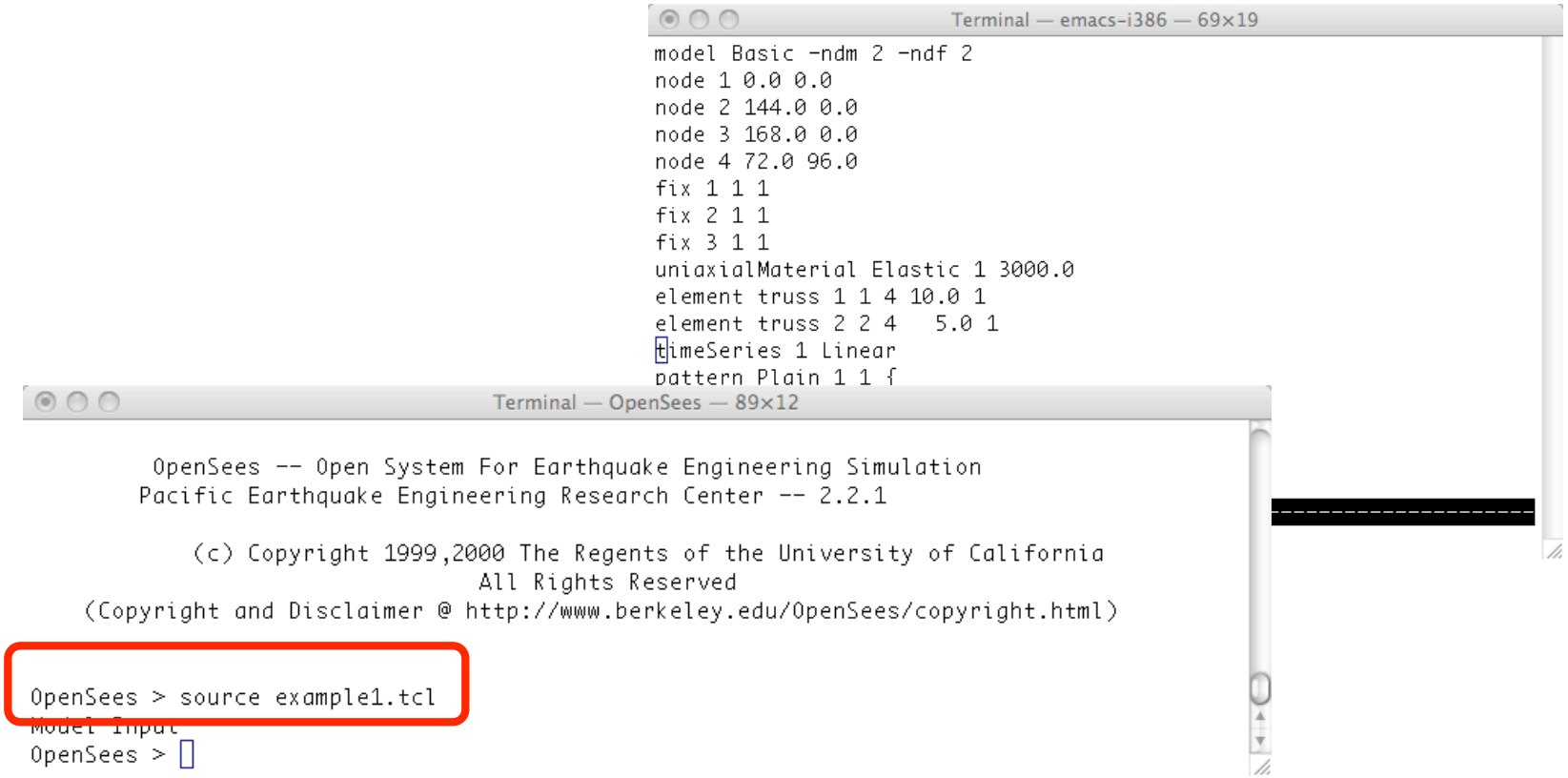
3 Ways to Execute the commands

1. Interactively - the commands as we have shown can be input directly at the prompt



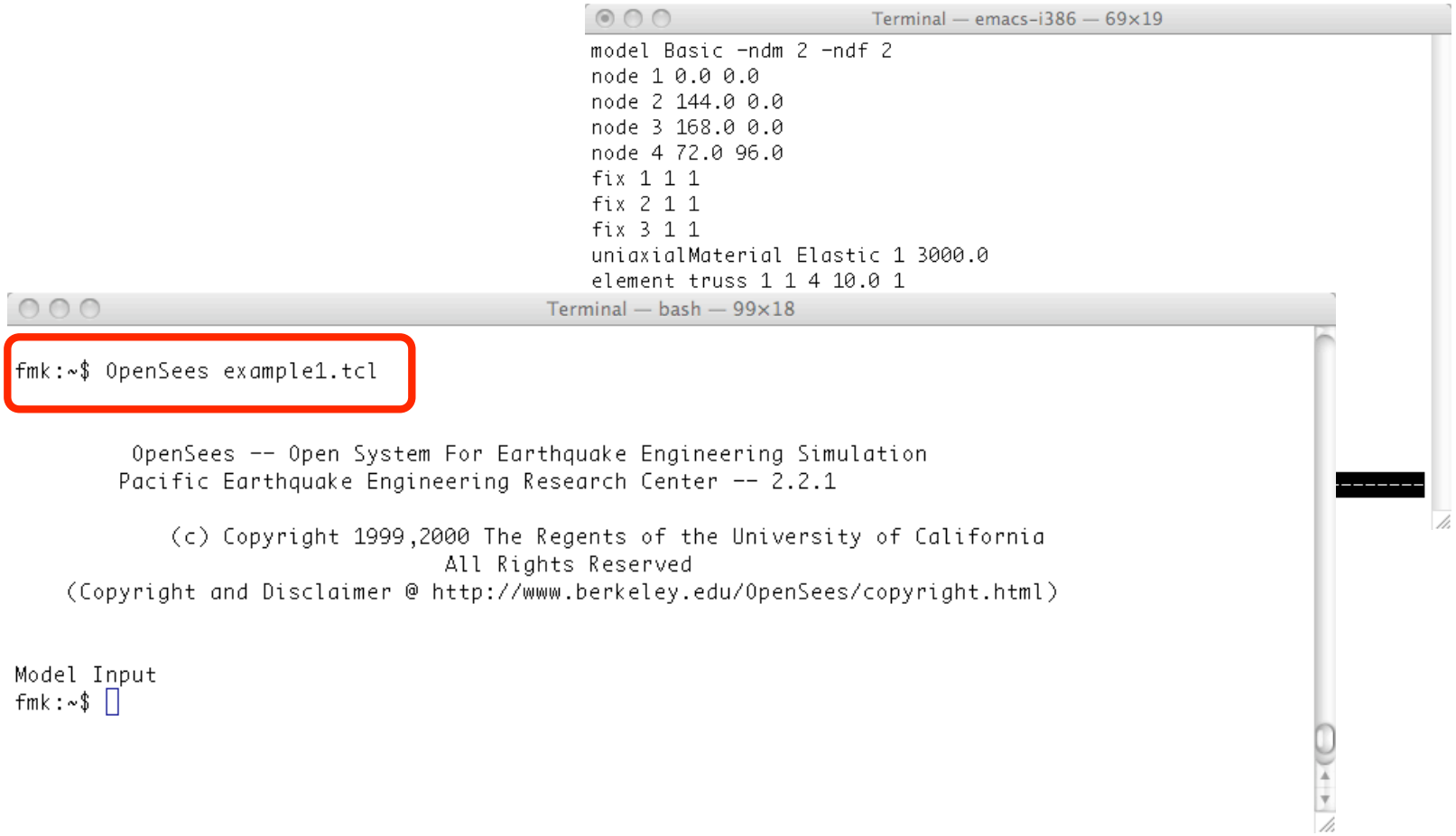
3 Ways to Execute the commands

2. Sourced from File- the commands are placed in a text file which is sourced in



3 Ways to Execute the commands

3. Batch Mode- the commands are placed in a text file which are executed at startup.



if batch mode - useful default variables: argv & argc

Terminal — emacs-i386 — 101×39

```
#parse input
if {$argc != 1} {
  puts "Incorrect Usage: OpenSees example2.tcl $E"
  exit
} else {
    set E [lindex $argv 0]
}

# model
model Basic -ndm 2 -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 $E
element truss 1 1 4 10.0 1
element truss 2 2 4   5.0 1
timeSeries Linear 1
pattern Plain 1 1 {
   load 4 100.0 -50.0
}

#analysis
integrator LoadControl 1.0
algorithm Linear
numberer Plain
constraints Plain
system BandGeneral
analysis Static
analyze 2

#output
puts "node 4 disp [nodeDisp 4]"
```

Terminal — bash — 101×39

```
fmk:~$ OpenSees example2.tcl 3000.0


        OpenSees -- Open System For Earthquake Engineering Simulation
       Pacific Earthquake Engineering Research Center -- 2.2.1

          (c) Copyright 1999,2000 The Regents of the University of California
                                All Rights Reserved
    (Copyright and Disclaimer @ http://www.berkeley.edu/OpenSees/copyright.html)


node 4 disp              1.8750000000000000000            -0.88541666666666662966
fmk:~$ OpenSees example2.tcl 6000.0


        OpenSees -- Open System For Earthquake Engineering Simulation
       Pacific Earthquake Engineering Research Center -- 2.2.1

          (c) Copyright 1999,2000 The Regents of the University of California
                                All Rights Reserved
    (Copyright and Disclaimer @ http://www.berkeley.edu/OpenSees/copyright.html)


node 4 disp              0.93750000000000000000           -0.44270833333333331483
fmk:~$
```

OpenSees & Matlab

- Calling matlab from an OpenSees script (mScript.m)

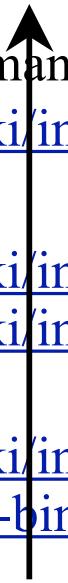
```
# invoke matlab
if {[catch {exec matlab -nosplash -nodesktop -r "mScript; quit"}]}
   {
 puts "Ignore this $msg"
}
```

- Calling OpenSees from a matlab script

```
 # invoke matlab
!OpenSees opsScript.tcl
```

OpenSees Resources
http://opensees.berkeley.edu

- Message Board - **look for answers**, **post questions** and **ANSWERS**
  http://opensees.berkely.edu/community/index.php
- Getting Started Manual - basic how to for getting started
  http://opensees.berkeley.edu/wiki/index.php/Getting_Started
- User Documentation - command documentation & theory!
  http://opensees.berkeley.edu/wiki/index.php/Command_Manual
- User Examples
  http://opensees.berkeley.edu/wiki/index.php/OpenSees_User
  http://opensees.berkeley.edu/wiki/index.php/Examples_Manual
- Developers
  http://opensees.berkeley.edu/wiki/index.php/OpenSees_Developer
  http://opensees.berkeley.edu/cgi-bin/cvsweb2.cgi/OpenSees/SRC/



OpenSeesWiki - Editable by all **(including YOU!)**

Any Questions?