Reliability/Sensitivity/Optimization-Enabling Technologies for Transportation Research Using OpenSees

Prof. Michael H. Scott (OSU) and
Prof. Kevin R. Mackie (UCF)

PEER Transportation Systems Research Program (TSRP)
Richmond Field Station
Richmond, CA

May 2, 2011

Project Objectives

1. Support PEER research in transportation systems
2. Improve usability of reliability and sensitivity modules
    - More flexibility in creating uncertainty models
    - Obtain uncertainty analysis results in free-form
3. Software redesign of uncertainty modeling framework
    - Reduce tight coupling of classes and code duplication
    - Profiling to identify computational bottlenecks
    - Extend framework beyond familiar application domains
4. Documentation of changes and example problems
    - OpenSees users workshop
    - Online documentation
5. Consistency with other OpenSees developers
    - SNOPT optimization at UCSD
    - TELM reliability at UCB

Improvements in Usability (*Completed*)

1. Allow non-sequential tagging of uncertainty objects
   - Parameters, random variables, positioners, performance functions
   - Free-form approach for uncertainty analysis of large models
   - Tcl commands return lists of tags for script-level data management
2. Identify and update parameters within a Tcl script
   - Easy approach to Monte Carlo simulation
   - Finite difference sensitivity without wiping out entire model
   - Avenues in to “black box” for third party FE model updating
3. Rethink how uncertainty analysis results are given to user
   - Fixed-format text output file –> house of cards
   - Tcl commands, e.g., `[getBetaFORM $pfTag]` will allow user to create files with customized format

Improvements in Usability (*Completed*)

4. Use Tcl API to evaluate performance functions
   - Parsing of specialized syntax was difficult to maintain/extend `performanceFunction 2 ‘‘0.2 - u_2_1’’` (*Now deprecated*)
   - Use Tcl's inherent variable substitution and procedure calls (user-defined or OpenSees-defined) `performanceFunction 2 ‘‘0.2 - \[nodeDisp 2 1\]’’`
5. User-defined gradients of performance functions
   - Reliance on finite differences even when analytic gradients of FE response
   - Simple performance functions: use OpenSees/Tcl commands `gradPerformanceFunction 2 $rvTag ‘‘0.2 - \[sensNodeDisp 2 1 $paramTag\]’’`
   - Use Tcl API for regular expressions to automatically translate `nodeDisp` to `sensNodeDisp` (*In progress*)

Changes to High Level Classes (*In progress*)

1. Overhaul of uncertainty analysis
   - Strip classes down to basic functionality
   - Derive FORM, SORM, etc. from base class
2. Uncertainty analysis results
   - Currently stored in performance function class
   - Use separate "observer" classes to store results
3. Pure virtual evaluator of performance function
   - Calls to Tcl_Eval mired in base class
   - Move to subclass for when other interpreters, e.g., Python, are linked with OpenSees
4. Four types of parameters derived from abstract Parameter class
   - FE model and its response
   - Random and design variable parameters
   - Uncertainty model and its response
   - User-defined scripting language parameters

Moving Forward

1. Tasks to be completed during Summer 2011
    - Documentation of new user commands and examples
    - Changes to high level classes
    - Documentation of software changes
2. Multiple developers, one source code repository
    - OpenSees is modular, but interfaces are intertwined
    - Synchronize with SVN frequently
    - Resolve source code conflicts when merging with repository
3. Other TSRP application areas
    - Combined live load and seismic analysis of bridges
    - Fluid-structure interaction for wave loading on bridges