Open System for Earthquake Engineering Simulation

Pacific Earthquake Engineering Research Center

PEER/NEESit OpenSees Developer Symposium

OpenSees Development Process and Discussion

Some OpenSees Kernel Development Needs

- "Input" application support
  - (Formal) scripting language
  - GUI interface
- "Output" application support
  - Formal output description, including metadata
  - Graphics, viz interface(s)
- Simulation support
  - Equation partitioning, DOF types
  - Parameterization of models
  - Inter-component communication
- Software architecture (primarily HPC)
  - Multi-threading
  - Memory management (localization and caching)

The Open Source “Movement”

- The objective of open source is to develop a community with a common interest, goals, and benefits from software
- Many examples of successful open source:
  - Linux
  - Apache
  - Mozilla
  - Sakai
- Problems with open source: longevity, reliability, forking, intellectual property.
- How to make this work for engineering software? For civil engineering software?

Key Issues

- Strategic issues
- Technical issues
- Sociological issues
- Legal issues

Technical Issues

- Framework support different simulation and analysis methods
- Computational efficiency
- Validation and (software) reliability
- Support for multi-platform computers and OS
- Libraries and configuration management
- Coding standards, namespaces, source documentation, etc.

Strategic Issues

- What is the target audience for OpenSees
- Support for NEES and NEESR applications

Sociological Issues

- Education
- Documentation
- Support
- Project roadmaps
- Credit for contributions
- Forking of code

Legal Issues

- Most OpenSees software is copyright by UC Regents. It is not an “open source license”.
- Contributions to OpenSees need contributor’s agreement, which can be non-exclusive. Universities or employers may have requirements for contributor’s agreement.

Community Based Software Development Process

- Examples:
  - Linux
  - Apache
  - Java Development Process
  - SourceForge projects
- Important aspects:
  - Communication
  - Prioritization
  - Reliability

Open Source Development Process

- Roadmap
  - What are needs?
  - What is near term, long term plan?
- Project management
  - Definition of a project
  - Propose/accept project?
  - Project status tracking

Open Source Development Process

- Review and acceptance process--extent depends on the level (kernel, trusted, general)
  - Design and API's with OpenSees
  - Code review
  - Documentation review
  - Example review
  - Validation and verification
  - Public comment
- Branch management
  - Development branches
  - Release trunk

Questions

Discussion

Action Plan