A framework to support distributed testing and service integration in earthquake engineering

Martin Williams and Ignacio Lamata
University of Oxford, UK
Contents

• Introduction
• Distributed hybrid testing
• Celestina – computing tools to promote collaboration
• Celestina-Sim framework for distributed hybrid testing
• Proof-of-concept tests between Oxford and Kassel
• Conclusions
Introduction

• Presentation relates to distributed hybrid testing but elements of it may be relevant to single-lab hybrid testing too

• Need for systematic approach, and a common language, to promote international collaboration, taking account of differing hardware, software, protocols in different labs

• Work performed by Ignacio Lamata at Oxford under the SERIES project, and during his later collaborative work with Shirley Dyke’s group at Purdue University

• Proof-of-concept tests conducted between Oxford and Kassel, with input of Uwe Dorka and Ferran Obon-Santacana
Distributed hybrid testing

- An extension of hybrid testing in which physical or numerical substructures are located in geographically remote labs
- Performed at extended timescales between Oxford-Bristol-Cambridge, and fast between Oxford-Bristol
Fast hybrid tests between Oxford-Bristol

Earthquake Engng Struct. Dyn.
doi: 10.1002/eqe.2385
Issues with distributed testing

- Increased complexity compared to single-laboratory testing
- Need to interface between different software, hardware, and operational procedures
- Difficulty of error-tracing
- Need for intensive human interaction prior to testing*
- Researchers engaged in tedious tasks that could be automated

doi: 10.1098/rsta.2010.0140
A suite of applications aimed at improving international collaboration through data-sharing and joint testing
Celestina-Sim*

• Divide activities into:
  – high level: managing and planning a test
  – low-level: running the test

• A specification to support high-level activities such as:
  – identification and location of participants
  – experimental planning
  – results collection

Celestina-Sim services overview

**Networking services:**
To know how to contact others and discover new managers and laboratory facilities.

**Definition services:**
To verify that the simulation can be successfully conducted: network links work, data types are understood, simulation dates are correct and the simulation plan is accepted.

**Testing services:**
To order the test execution according to the test phases, to collect results and to abort a test when errors occur.

Actual data exchange
Networking services

- Peer-to-peer (P2P) network operated above the Internet infrastructure
- Any machine can access any other without intermediaries
- High-level *Sky nodes* – managers in charge of controlling the experimental plan
- Low-level *Ground nodes* – in charge of simulation execution
Definition services

• To verify that a test is feasible
• Verifications commanded by a sky node and executed by a ground node
• Main items are:
  – Network link – verification that the appropriate links can be established between participants
  – Data compatibility – check that each participant can correctly read and understand the others’ data and commands
  – Simulation plan agreement – agree participants, data to be exchanged, simulation workflow, speed of test execution etc.
Testing services

- Sky node manages test by setting state of all other nodes
  - Initially Available
  - Move to Not Ready while preliminary testing tasks are performed, then to Ready
  - Return to Not Ready when main simulation phase starts
  - At the end of the test, Sky node sends abort commands, returning all nodes to Available
Implementation example

- A purely numerical simulation of an earlier, local hybrid test
- 4DOF model of third-stage separation of Arianne IV rocket launch
- Linked simulations at Oxford and Kassel
Implementation example

- Arrangement of sky and ground nodes for the test
Execution of the simulation
Sample results

- Comparison of the original hybrid test, a single-site numerical simulation, and a distributed simulation using Celestina:

<table>
<thead>
<tr>
<th></th>
<th>Celestina (Fast network TCP)</th>
<th>Direct communication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum ms/step</td>
<td>25.99</td>
<td>25.67</td>
</tr>
<tr>
<td>Average ms/step</td>
<td>28.82</td>
<td>28.06</td>
</tr>
</tbody>
</table>
Conclusions

• Celestina-Sim provides a framework for distributed simulation, enabling heterogeneous systems to collaborate in a systematic way
• It is a specification rather than a specific piece of software. However, the easiest way to implement it is to re-use or adapt the Java implementation developed by Lamata
• Steps will be taken to publish the Celestina framework under an open-source license that allows institutions to use and adapt the framework