OpenFresco Commands in an Example

KHALID M. MOSALAM
& SELIM GÜNAY
UNIVERSITY OF CALIFORNIA, BERKELEY
Communication between the experimental and analytical substructures is obtained in OpenFresco with the following commands (From another perspective: Experimental substructure is introduced to the finite element model with the following)

Reminder: There are other options for the defined experimental properties, for example `truss` for `expElement`, `ActorSite` for `expSite`, `ThreeActuators` for `expSetup` and `xPCtarget` for `expControl`
Communication between the experimental and analytical substructures is obtained in OpenFresco with the following commands (From another perspective: Experimental substructure is introduced to the finite element model with the following)

```
# Define experimental element
expElement beamColumn expID $expNI $expNO 1-site -initStif $Es 0 0 0 $kexpcol 0.0 0 0.0 0.0;
```

- **expElement**: defines the experimental element. After this definition, the finite element model knows:
  1. Tag of the experimental element
  2. The nodes where the element is connected to (connectivity of the element)
  3. **Orientation** of the element and the presence of second order consideration (No, p-delta or co-rotational)
  4. The tag of the **experimental site** where the element is located at
  5. **Initial stiffness** of the element (required for implicit integration)

**Remark**: Experimental element in this example contains only axial and lateral stiffness terms since there is no moment transfer at the top of the column
Communication between the experimental and analytical substructures is obtained in OpenFresco with the following commands (From another perspective: Experimental substructure is introduced to the finite element model with the following):

```
# Define experimental site
expSite LocalSite 1 1;
```

expSite: defines the experimental site. After this definition, the finite element model knows:
1) Tag of the experimental site
2) Tag of the experimental setup used at the experimental site
Communication between the experimental and analytical substructures is obtained in OpenFresco with the following commands (From another perspective: Experimental substructure is introduced to the finite element model with the following)

```
# Define experimental setup
expSetup OneActuator 1-contol 1- sizeTrialOut 3 3;
```

expSetup: defines the experimental setup. After this definition, the finite element model knows:

1) Tag of the experimental setup
2) Tag of the experimental control used with experimental setup
3) Direction of the quantity imposed to the actuator in the element's basic reference system
4) Size of the trial/output vector received from/returned to the element: These should be compatible with the size of the initial stiffness matrix. For example, if the initial stiffness matrix is $3 \times 3$, then the size of trial and output vectors should be 3
Communication between the experimental and analytical substructures is obtained in OpenFresco with the following commands (From another perspective: Experimental substructure is introduced to the finite element model with the following)

```
# Define Experimental Control
expControl SCRAMNet 1 381020 3
```

expSetup: defines the experimental setup. After this definition, the finite element model knows:
1) Tag of the experimental control
2) Memory offset from SCRAMNet base memory address [bytes]
3) Number of actuator channels in the control system

SCRAMNet control:
- Guarantees continuous movement of actuators by providing a continuous communication of command and feedback between the experimental element and the computational platform
- SCRAMNet memory avoids any delays between the moment that the computation is completed and the moment this information is sent to the controller

Generally don’t change from test to test in an established lab system
Example: Distributed HS

Experimental Substructure: in Berkeley
Analytical Substructure: in Minho

tcl file in Minho: RegularOpenSees files with two additional commands

```plaintext
loadPackage OpenFresco

# Define experimental element
expElement beamColumn $expID $expNI $expNJ 1 -site 1 -initStif $Es 0 0 0 $kexpcol 0.0 0 0.0 0.0

# Define experimental site
expSite ShadowSite 1 "169.229.203.152" 8091
```

expElement: same as in the case of the local file
expSite: After this definition, the finite element model knows that the experimental element is at an experimental site with the input IP address and port *
**Example: Distributed HS**

Experimental Substructure: in Berkeley

Analytical Substructure: in Minho

---

**tcl file in Berkeley: OpenFresco file**

```
# Define experimental site
expSite ActorSite 1 -setup 1 8090

# Define experimental setup
expSetup OneActuator 1 -control 1 2 -sizeTrialOut 3 3;

# Define Experimental Control
expControl SCRAMNet 1 381020 8
```

expSetup and expControl: same as in the case of the local file

expSite: ActorSite replaces the Shadowsite in the computational platform. The site information required in the computational platform is obtained from the Actorsite, such as the experimental setup used in the site.