LoadGeneratorModel.java

**Summary:** This module generates tasks in a list of mobile devices based on the virtual start time & following nine simulation scenarios.

1. HAFA\_ORCHESTRATOR,
2. CENTRALIZED\_ORCHESTRATOR,
3. LOCAL\_ONLY,
4. CLOUD\_ONLY,
5. EDGE\_BY\_LATENCY,
6. EDGE\_BY\_DISTANCE,
7. FIXED\_NODE,
8. SELECTED\_LEVELS,
9. SELECTED\_NODES
10. List<EdgeTask> getTaskList()

This method lists all the tasks. Virtual start time is used while generating the task.

1. int getNumberOfMobileDevices()

It gets the nbr of mobile devices.

1. void setNumberOfMobileDevices(int numberOfMobileDevices)

It sets the nbr of mobile devices.

1. double getSimulationTime()

It gets simulationTime

1. void setSimulationTime(double simulationTime)

It sets the simulationTime.

1. String getSimScenario()

It gets SimScenario

1. void setSimScenario(String simScenario)

It sets SimScenenio

IdleActiveLoadGeneratorModel.java

**Summary:** This module generates exp nbr for I-O file size and task length. It also generates tasklist using --

different APP\_TYPES(i), randomTaskType, virtualTime, wifi, sensor, actuator & expRngList[i][j] where i=0 thru APP\_TYPES and j=5,6,7

initializeModel()

ExponentialDistribution[][] expRngList = new ExponentialDistribution[APP\_TYPES][3]; APP\_TYPES = 7;

for(int i=0; i<7; i++) {

creates expRngList[i][0], expRngList[i][1], expRngList[i][2]

from ExponentialDistribution(); getRandomSeed(); getTaskLookUpTable()[i][j] j = 5,6,7 //why ??;

}

for(int i=0; i< numberOfMobileDevices; i++) {

Creates poissonMean, activePeriod, idlePeriod, From getTaskLookUpTable()[TaskType][k] k = 2,3,4

Creates activePeriodStartTime from getRandomDoubleNumber(10, 10+activePeriod); //start from 10th sec

Creates virtualTime from activePeriodStartTime; //start from 10th seconds

boolean sens = (SimUtils.getRandomDoubleNumber(0, 100) < 67);

boolean act = (!sens || SimUtils.getRandomDoubleNumber(0, 100) < 50); boolean wifi = true;

EdgeTask newTask = new EdgeTask(i, randomTaskType, virtualTime, expRngList, wifi, sens, act);

taskList.add(newTask); }