RAVEN Statistical Framework

RAVEN Entities and Input Structure introduction

Idaho National Laboratory - Idaho Falls, ID

www.inl.gov

Idaho National Laboratory

Objectives

- Learn the concept of “Entities” in the RAVEN framework
- Learn how these “Entities” are implemented in RAVEN
- Learn the concept of RAVEN “Step”
- Learn how RAVEN Steps and Entities are assembled in the input file for the construction of a calculation flow: From “Entities” to “Actors”

- Basically, you should be able to start playing with RAVEN

- Additional info
  - RAVEN user manual and user guide
  - Input files shown in this workshop
  - RAVEN regression and analytical tests

Entities in RAVEN?

- An *Entity* is a category of objects aimed to employ a certain action
- All the objects that belong to a certain *Entity* have a common definition of their Input/Output needs

Calculation control

RunInfo

- Number of simultaneous Model evaluations
- Execution sequence
- …

Physical/Mathematical Modeling

Models

- Codes
- Surrogate Models
- Post-Processors
- …

IO by files

Files

- Codes' input files
- Post-Processor outputs
- …

Storage

Databases

DataObjects

- Codes' outputs
- RAVEN solutions
- …

Stochastic Modeling

Distributions

- 1D distributions
- ND distributions

Perturbation

Samplers

Optimizers

- Uncertainty propagation
- Parameter optimization
- …

Export solutions

OutStreams

- CSV/XML output
- Plotting
- …

User input functions

Functions

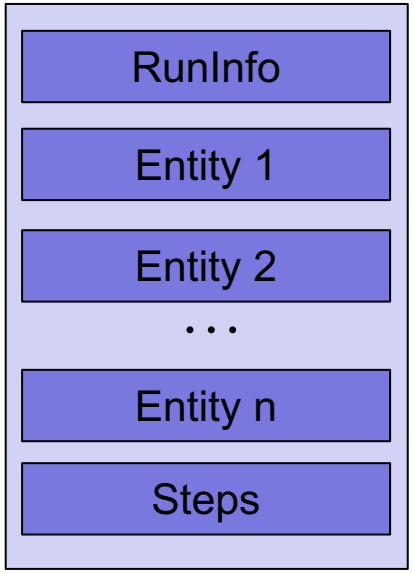
- User defined functions
- Goal functions
- …

The Analysis: the RAVEN Approach

- Type of information
  - Desired stochastic analysis
    - What do I want to do? ← RunInfo
  - Entities needed
    - What do I want to use? ← Entities
    - How do I want to use them? ← Steps

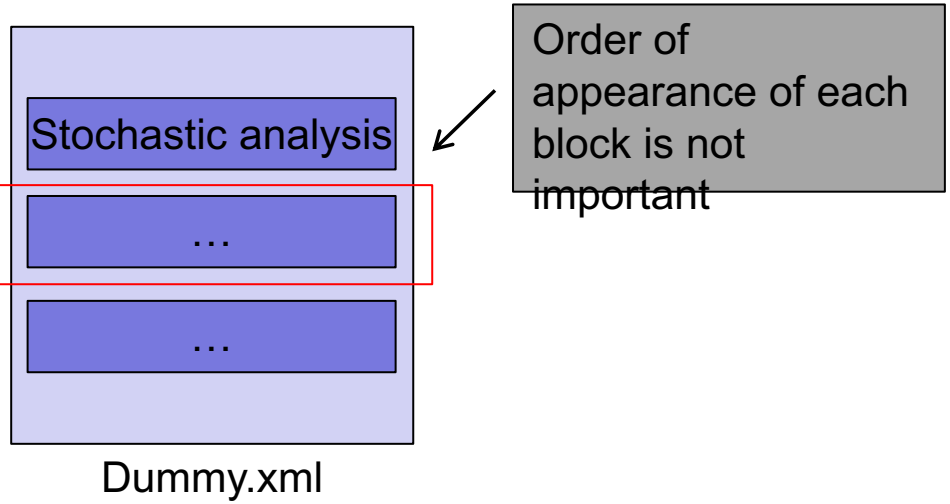
Raven semantics

- Template of RAVEN input file



Analysis Input File: the RAVEN Approach

- One single input file
  - High modular input style



  - .xml format

Node

Attribute

```
<Samplers>
    <Stratified name='test'>
        <samplerInit>
            <seed>1234</seed>
        </samplerInit>
    </Stratified >
</Samplers>
```

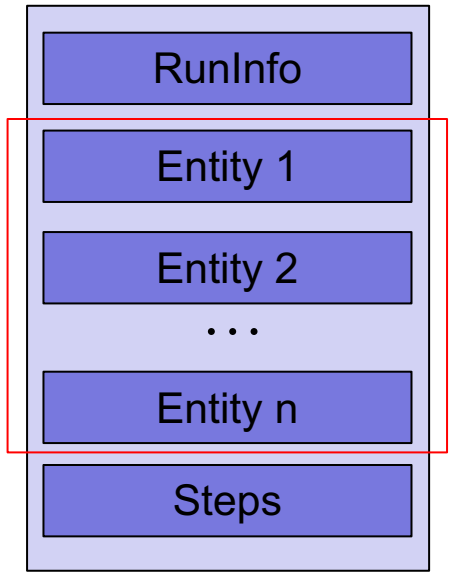
Sub-node

Construct the required Entities of an analysis

Entities Implementation and Input

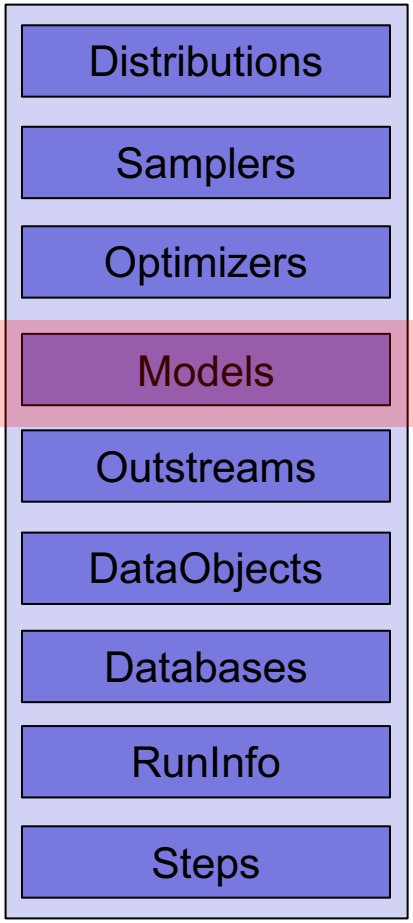
Entity: Models

- Models: projection from input to output space
  - Codes: through code interfaces
  - External models: python based module
  - Reduced Order Models (ROMs)
  - HybridModels: blend of ROMs and High-Fidelity
  - PostProcessors: used to perform action on data
    - Basic statistic operations
    - Comparison statistic
    - Clustering and data mining
    - ….
  - EnsembleModel: combination of multiple models



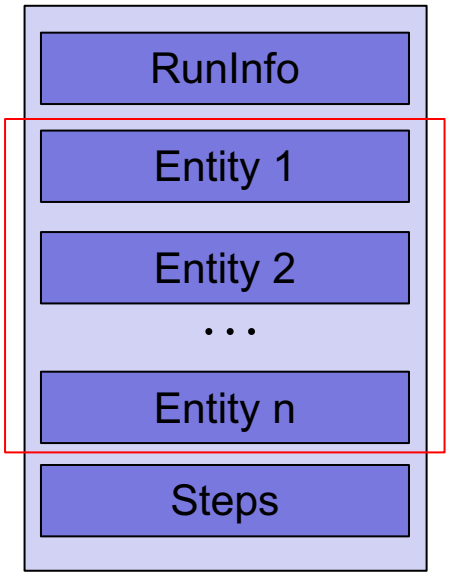
Input Structure: Models

```
<Models>
  <ExternalModel name='AName1' subType=''
    ModuleToLoad='./externalModel'>
    <variables>x,y,z</variables>
  </ExternalModel>
  <Code name='AName2' subType='Relap5'>
    <executable>path/to/executable</executable>
  </Code>
  <ROM name='AName3' subType='NDinvDistWeight'>
    <Features>x,y</Features>
    <Target>z</Target>
    <p>3</p>
  </ROM>
  <PostProcessor name='AName4' subType='BasicStatistics'>
    <expectedValue prefix='mean'>z,x,y</expectedValue>
    <variance prefix='var'>z,x,y</variance>
    <sensitivity prefix='sens'>
      <targets>z</targets>
      <features>x,y</features>
    </sensitivity>
  </PostProcessor >
</Models>
```



Entity: DataObjects (1/2)

- DataObjects: how data is stored within RAVEN
  - Format: (input parameters, output parameters)
  - Point Sets
    - Each realization produces a single value for each response



a	b
1.2	4.5

Model

x	y
1.24	5

  - History Sets
    - Each realization results in a history of values

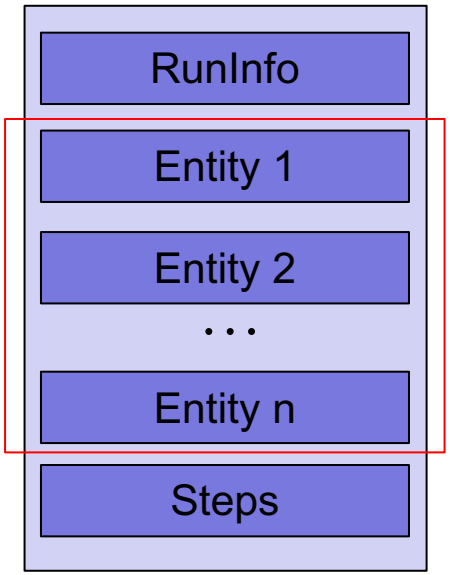
a	b
1.2	4.5

Model

t	x	y
0	0.04	13
1	0.36	8
2	1.24	5

Entity: DataObjects (2/2)

- DataObjects: how data is stored within RAVEN
  - Format: (input parameters, output parameters)
  - Data Sets (available in *EnsembleModel* only)
    - Each realization (e.g. scalar, vectors, etc.) produces a single value, vector or matrix for each response



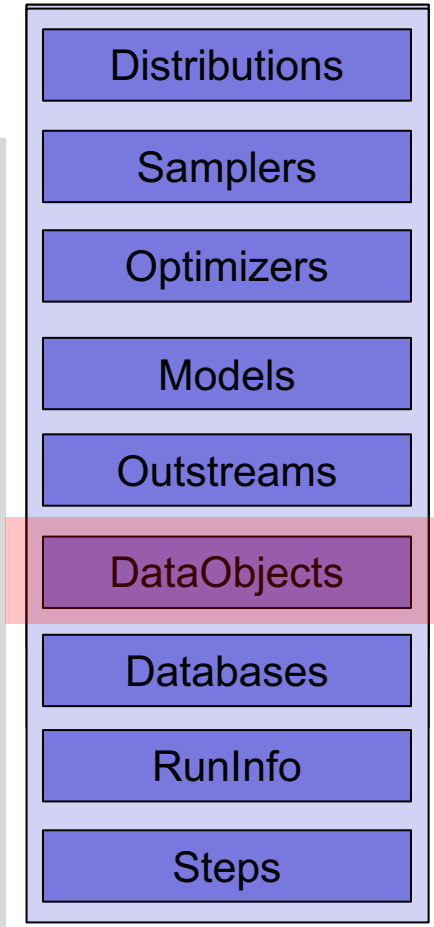
a(-)	b(bu)	bu
1.2	4.5	1
	5.0	2
	2.2	3

Model

t	x(t)	y(-)
0	0.04	13
1	0.36	
2	1.24	

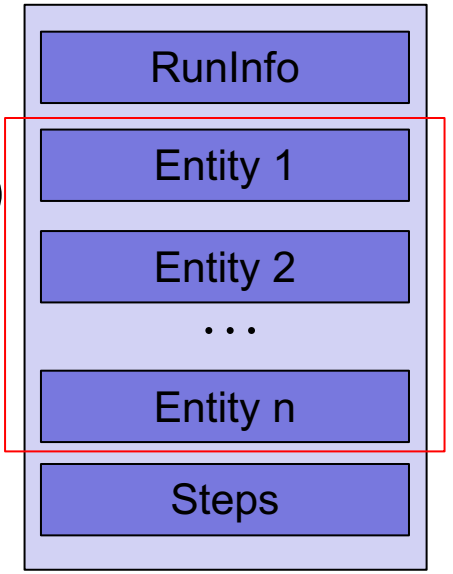
Input Structure: DataObjects

```
<DataObjects>
  <PointSet name='staticData'>
    <options>
      <operator>max</operator>
    </options>
    <Input>x,y,z</Input>
    <Output>o1,o2</Output>
  </PointSet >

  <HistorySet name='timedepData'>
    <Input>x,y,z</Input>
    <Output>o1,o2</Output>
  </HistorySet>

  <DataSet name='dataSetData'>
    <Input>x,y,z</Input>
    <Output>o1,o2</Output>
    <Index name='t'>o1</Index>
    <Index name='bu'>x,y</Index>
  </DataSet>
</DataObjects>
```



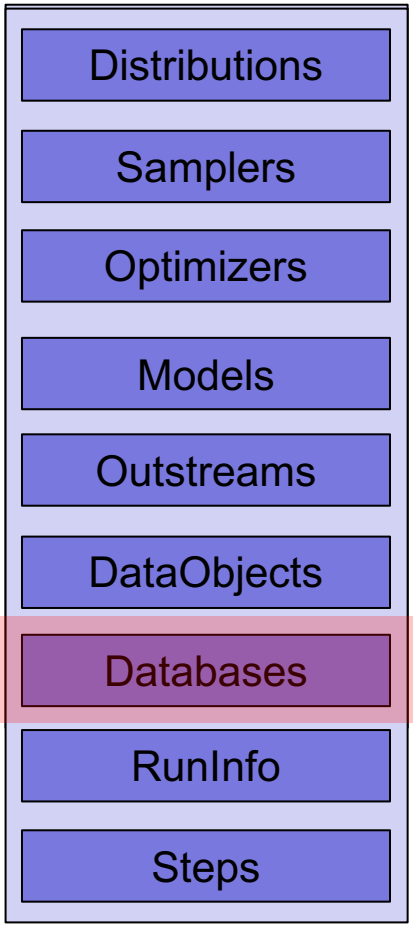
Entity: Databases

- Databases: data storage entities
  - Store data in binary format
    - Uncompressed (high memory–high performance)
    - Compressed (low memory-low performance)
  - HDF5 files
  - *DataObjects* can be saved into Databases
  - Existing Databases can be loaded into the RAVEN framework



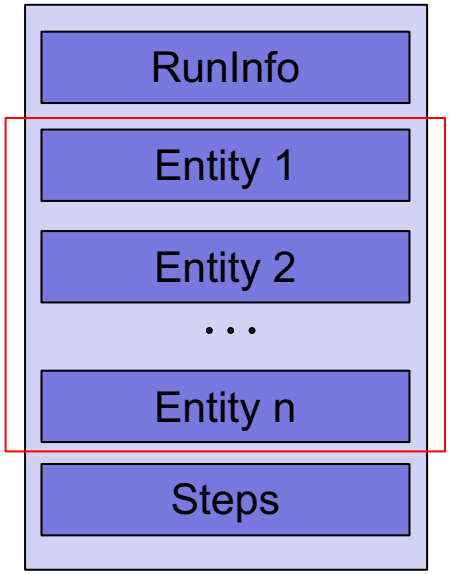
Input Structure: Databases

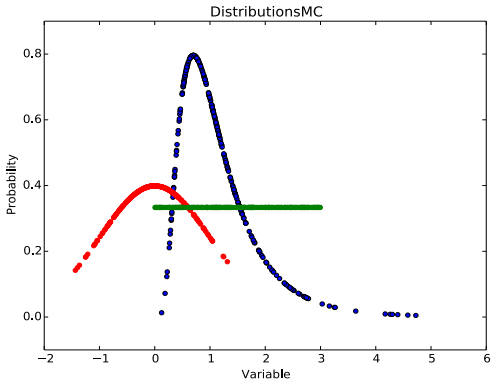
```
<Databases>
  <HDF5
    name='new_database_default_location'
    readMode='overwrite'
  />
 <HDF5
    name='new_database_directory'
    directory='path/to/desired/directory'
    readMode='overwrite'
  />
 <HDF5
    name='old_database'
    directory='path/to/desired/directory'
    readMode='read'
  />
</Databases>
```

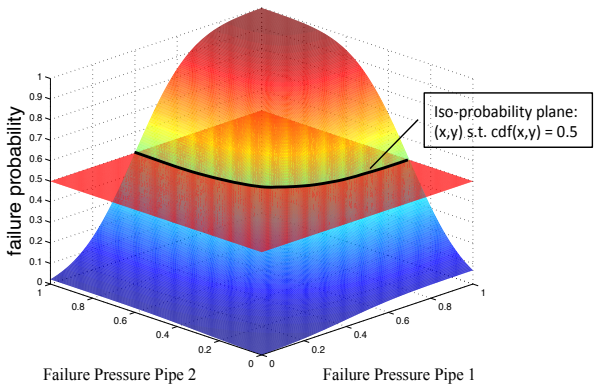


Entity: Distributions

- Distributions: stochastic representation of variable
  - 1D continuous and discrete:
    - Normal
    - LogNormal
    - Beta
    - Binomial
    - …
  - ND multi-dimensional distributions:
    - Multivariate Normal
    - Custom N-Dimensional

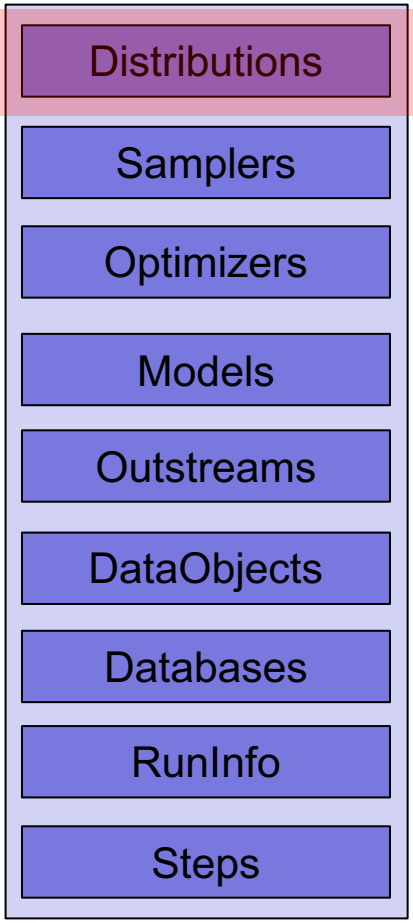






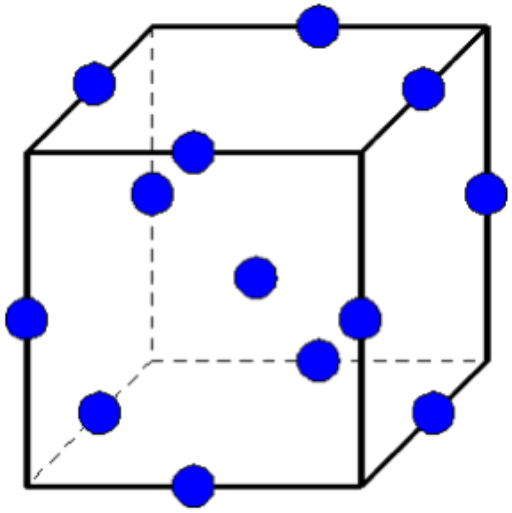
Input Structure: Distributions

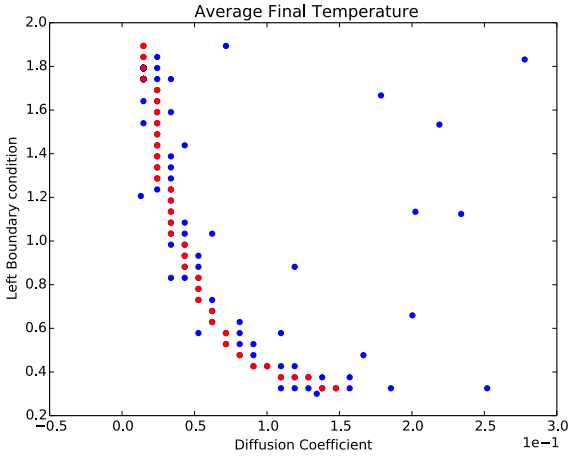
```
<Distributions>

  <Normal name='x_distrib'>
    <mean>2</mean>
    <sigma>0.2</sigma>
  </Normal>

 <MultivariateNormal name='y_z_distrib'>
    <mu>0.0 2.0</mu>
    <covariance>
     1.0   0.8
     0.8   1.0
    </covariance>
 </MultivariateNormal>

</Distributions>
```

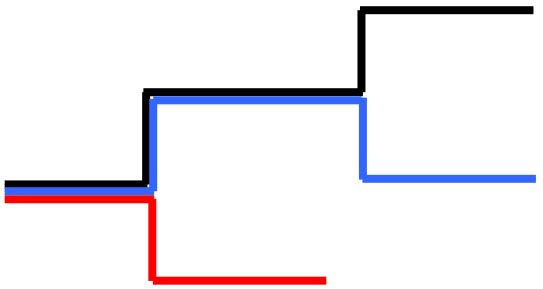


Entity: Samplers

- Samplers: input space sampling entities
  - Forward Samplers: Monte-Carlo, Stratified (LHS), Grid, Response Surface, Factorial Design, etc…
  - Adaptive Samplers (smart sampling)
  - Dynamic Event Tree Samplers

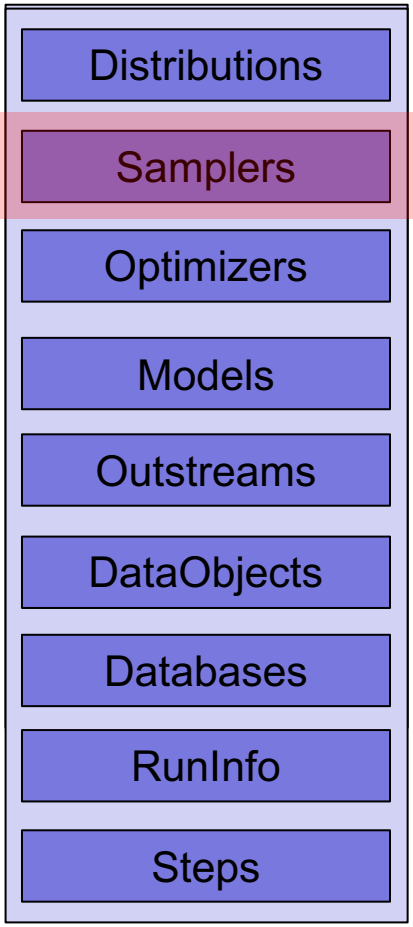






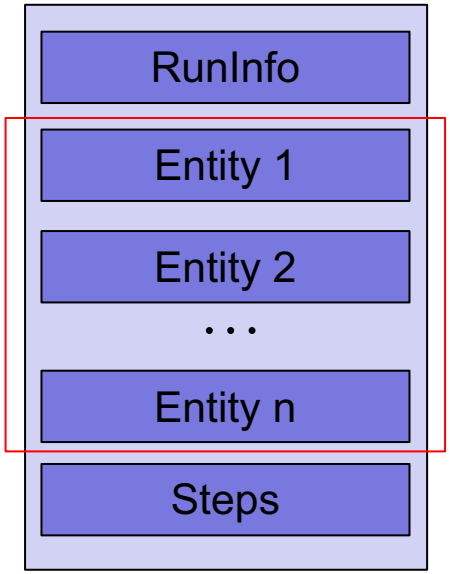
Input Structure: Samplers

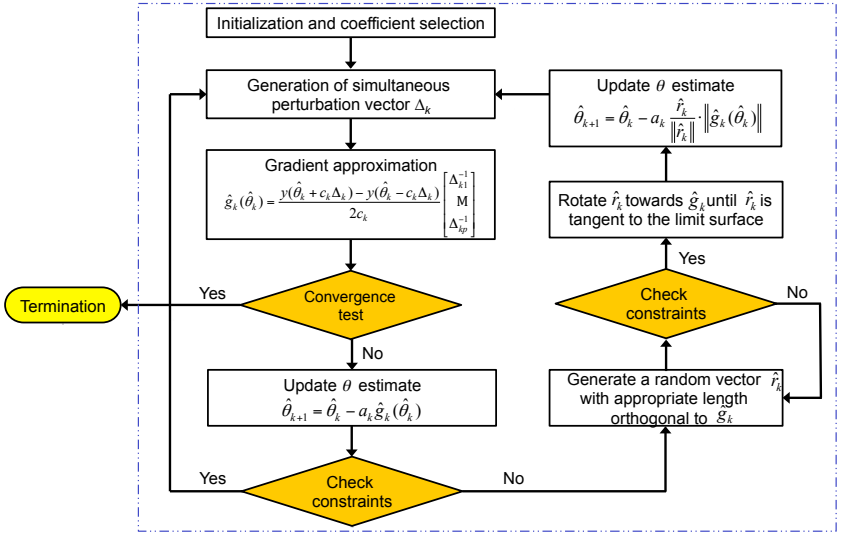
```
<Samplers>
  <MonteCarlo name='MCsampler'>
    <samplerInit>
      <limit>5000</limit>
    </samplerInit>
    <variable name='x'>
      <distribution>x_distrib</distribution>
    </variable>
  </MonteCarlo>
  <Grid name='GridSampler'>
    <variable name='x1'>
      <distribution>x1_distrib</distribution>
      <grid type='CDF' construction='equal' steps='10'>
        0.0 1.0
      </grid>
    </variable>
    <variable name='x2'>
      <distribution>x2_distrib</distribution>
      <grid type='value' construction='equal' steps='8'>
        0.1 0.9
      </grid>
    </variable>
  </Grid>
</Samplers>
```



Entity: Optimizers

- Optimizers: input space parameter optimization
  - Simultaneous Perturbation Stochastic Approximation (SPSA)
  - Finite Difference Gradient Descent
  - Genetic Algorithm-based optimizers*

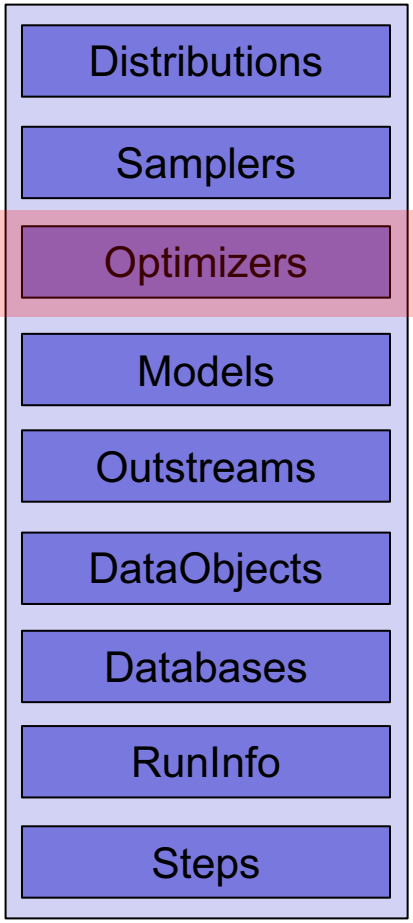




* planned to be developed

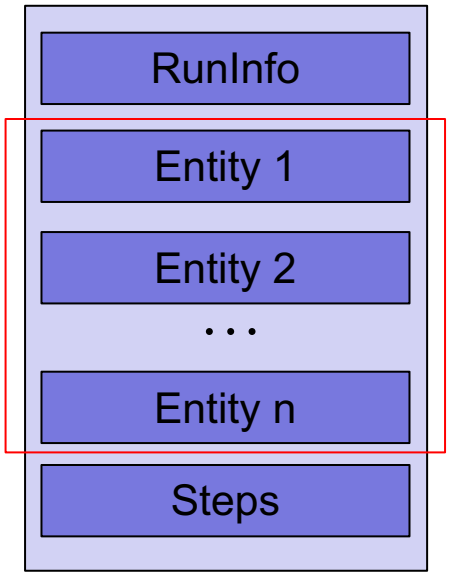
Input Structure: Optimizers

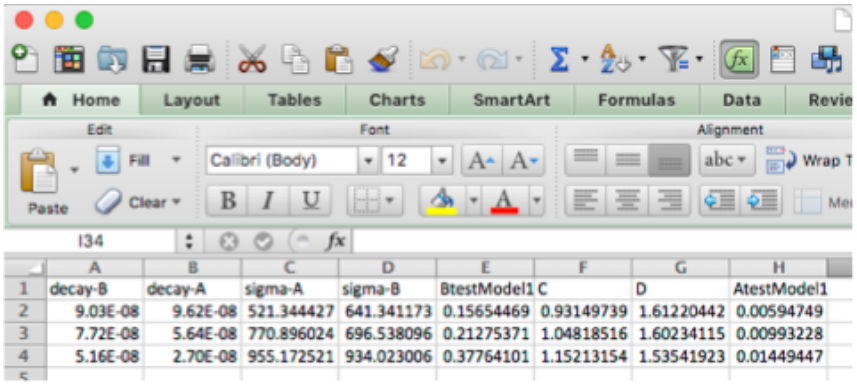
```
<Optimizers>
  <SPSA name='optmizer'>
    <initialization>
      <initialSeed>20021986</initialSeed>
    </initialization>
    <convergence>
      <gradientThreshold>1e-2</gradientThreshold>
      <gainGrowthFactor>2</gainGrowthFactor>
    </convergence>

    <TargetEvaluation name='DataObjects' type='PointSet'>
      optimizationOutput
    </TargetEvaluation>

    <variable name='x'>
      <lowerBound>0</lowerBound>
      <upperBound>10</upperBound>
      <initial>3.0</upperBound>
    </variable>

    <objectVar>outputToOptimize</objectVar>

  </SPSA >
</Optimizers>
```



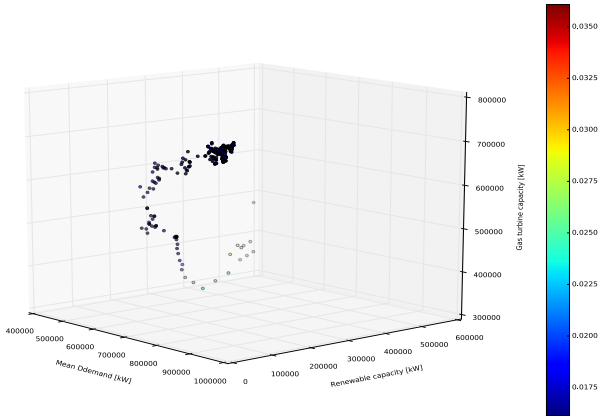
Entity: OutStreams

- OutStreams: used for data exporting/dumping
  - Printing:
    - DataObjects
    - Reduced Order Models (ROMs)
  - Plotting: both 2D and 3D plotting available
    - 4D by using color mapping
    - 5D by using marker size



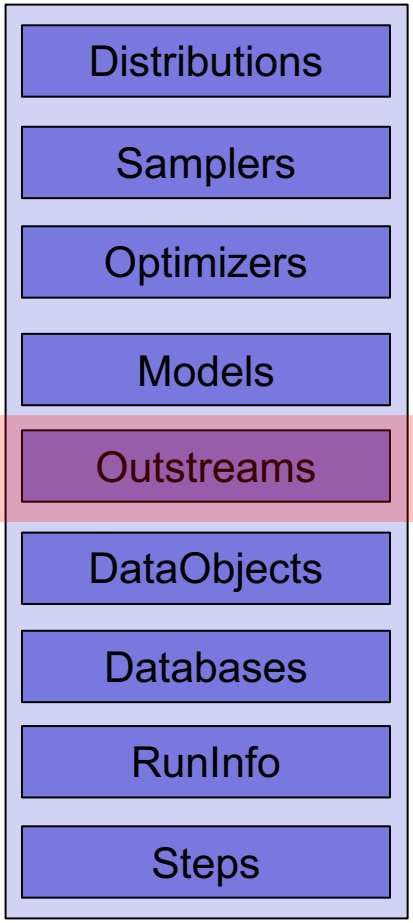


	A	B	C	D	E	F	G	H
1	decay-B	decay-A	sigma-A	sigma-B	BtestModel1	C	D	AtestModel1
2	9.03E-08	9.62E-08	521.344427	641.341173	0.15654469	0.93149739	1.61220442	0.00594749
3	7.72E-08	5.64E-08	770.896024	696.538096	0.21275371	1.04818516	1.60234115	0.00993228
4	5.16E-08	2.70E-08	955.172521	934.023006	0.37764101	1.15213154	1.53541923	0.01449447



Input Structure: OutStreams

```
<OutStreams>
  <Print name='exportCSV'>
    <type> csv </type>
    <source>ADataObjectOrROM</source>
  </Print>
  <Plot name='makeAPlot'>
    <plotSettings>
      <plot>
       <type> scatter </type>
       <x> ADataObject|Output|time </x>
       <y> ADataObject|Output|anOutput </y>
      </plot>
      <xlabel> time <xlabel>
      <ylabel> anOutput <ylabel>
    </plotSettings>
    <actions>
     <how> png, screen </how>
     <title> <text> This is a plot </text> </title>
    </actions>
  </Plot >
</OutStreams>
```

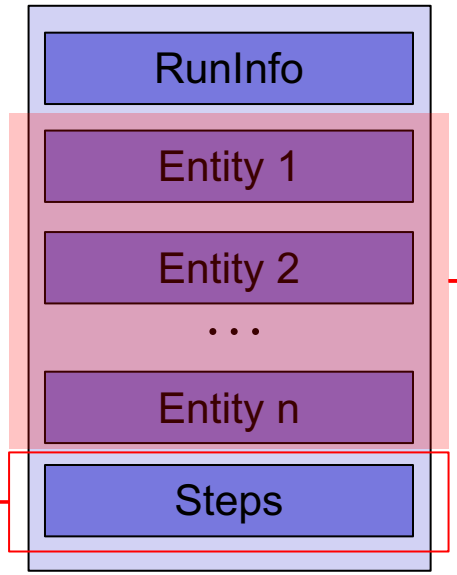


Construct the analysis flow

Steps

Steps

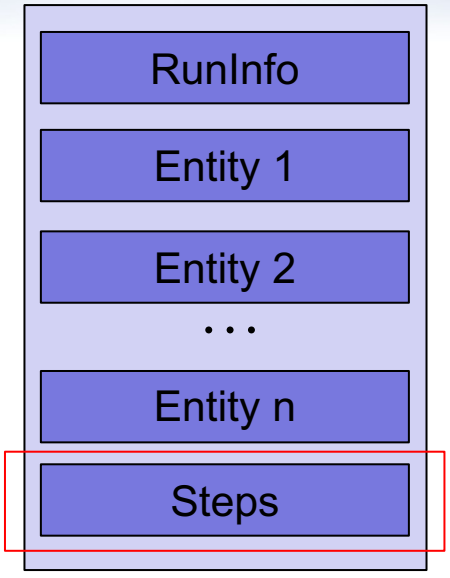
- A Step links Entities together to perform an action
- Multiple heterogeneous Entities are used in a single Step (DataObjects, Samplers, Models, …)
- All these Entities must be defined in their corresponding block
  - They can be defined after the Steps block




```
<Steps>
    <StepType1 name='simple_MultiRun'>
        …
    </StepType1>
    <StepType2 name='simple_PostProcess'>
        …
    </StepType2>
</Steps>
```

Steps

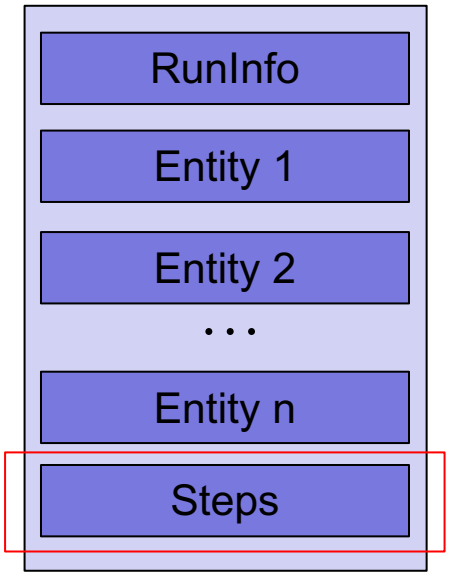
- Each Entity has a role
  - Input
  - Output
  - Model
  - Sampler
  - Function
  - ROM
  - Solution export




```
<Steps>
  ...
  <SingleRun name='StepName'>
    <Input  class='Files'       type=''    >anInputFile.i</Input>
    <Input  class='Files'       type=''    >anAuxiliaryFile</Input>
    <Model  class='Models'      type='Code'>aCode</Model>
    <Output class='Databases'   type='HDF5'>aDatabase</Output>
    <Output class='DataObjects' type='History'>aData</Output>
  </SingleRun>
</Steps>
```

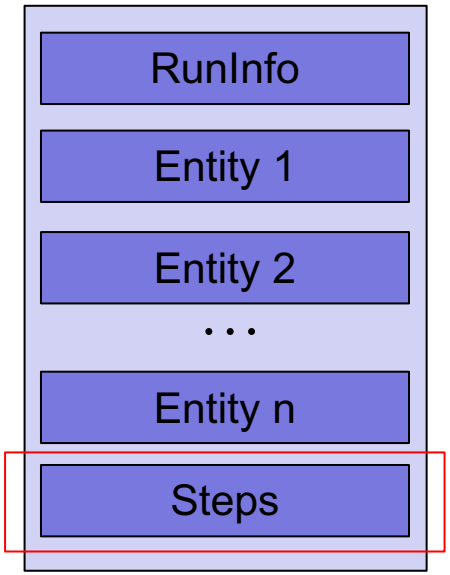
Step Types (1/2)

- SingleRun: perform a single run of a model
- MultiRun: perform multiple runs of a model
- RomTrainer: perform the training of a Reduced Order Model (ROM)
- PostProcess: post-process data or manipulate RAVEN entities



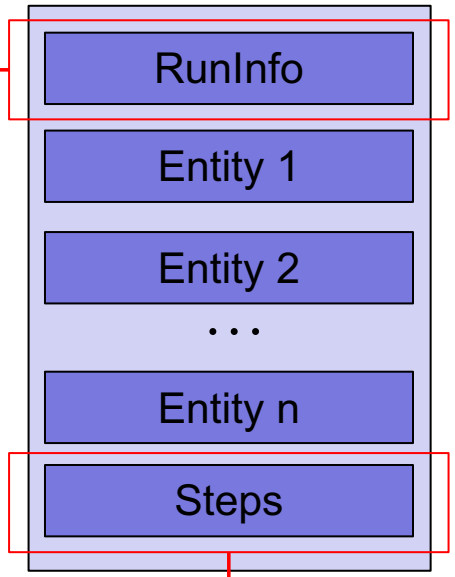
Step Types (2/2)

- IOStep:
  - construct/update a Database from a DataObjects and vice versa
  - construct/update a Database or a DataObjects object from CSV files
  - stream the content of a Database or a DataObjects out through an OutStream
  - store/retrieve a ROM to/from an external File using Pickle module of Python



RunInfo

- Desired analysis
  - Sequence of Steps
  - Working directory
  - Parallel computation parameters
  - …



# of simultaneous code evaluations

```
<RunInfo>
    <WorkingDir>./myDir</WorkingDir>
    <batchSize>6</batchSize>
    <Sequence>Step2,Step3,Step6</Sequence>
</RunInfo>
```

Thank you