

Introduction to Mosaik

TV. Jensen¹

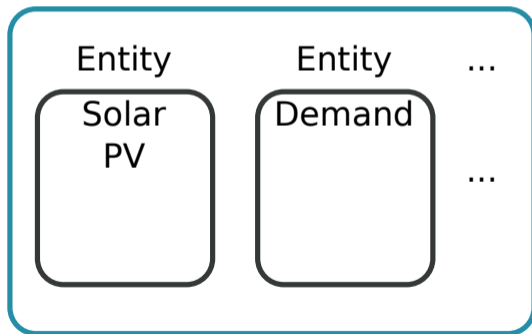
¹Department of Electrical Engineering
Technical University of Denmark

Structure of a Mosaik scenario

Mosaik couples a series of simulators

Each physical unit is represented by an *entity* in these simulators

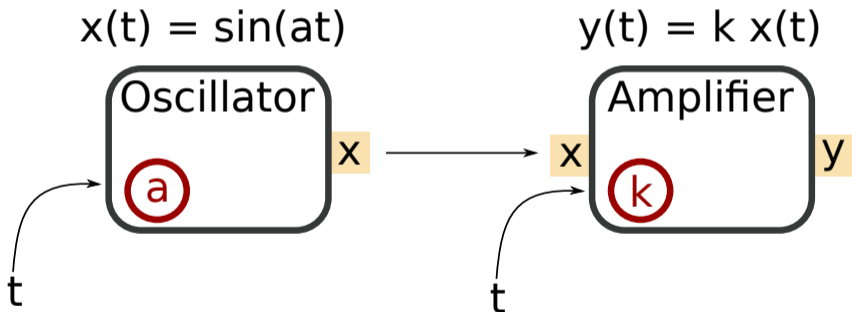
Simulator



Structure of a Mosaik scenario

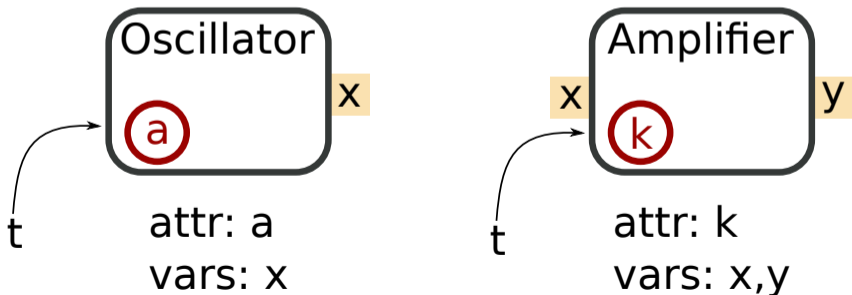
Attributes are set at initialization

Variables change over the course of the simulation



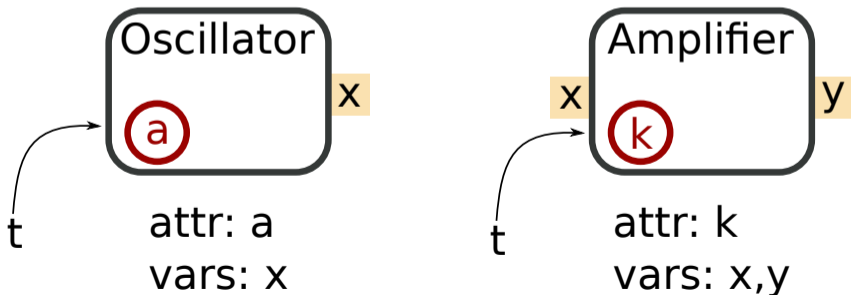
How to build Mosaik scenarios

1. Build simulators and entities (Python classes)



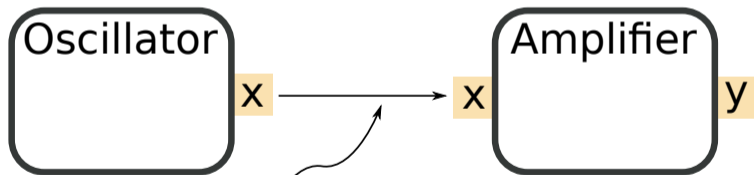
How to build Mosaik scenarios

1. Build simulators and entities (Python classes)
2. Instantiate a world, simulators and entities



How to build Mosaik scenarios

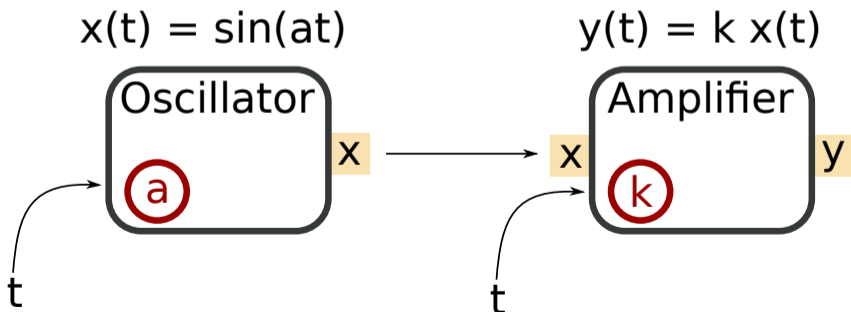
1. Build simulators and entities (Python classes)
2. Instantiate a world, simulators and entities
3. Connect entities



```
connect(Oscillator, Amplifier, ('x', 'x'))
```

How to build Mosaik scenarios

1. Build simulators and entities (Python classes)
2. Instantiate a world, simulators and entities
3. Connect entities
4. Run simulation



Mosaik Quirks

Mosaik has several quirks that make working with it easier.

BUT these quirks also restrict to some degree what you can do in Mosaik.

Mosaik Quirk # 1: Discrete time

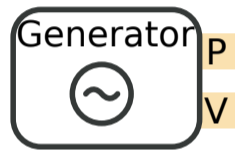
All Mosaik time steps are *discrete*. You have to handle how these discrete steps map to time steps.

MosaikTime	Simulation Time
1	00:00
2	00:10
3	00:20
4	00:30

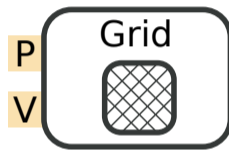
Typically, specify, e.g. 1 MosaikTime = 1 second, and adjust accordingly.

Mosaik Quirk # 2: No algebraic closure

Droop-controlled generator and weak link grid:



$$P = P_0 + kV$$

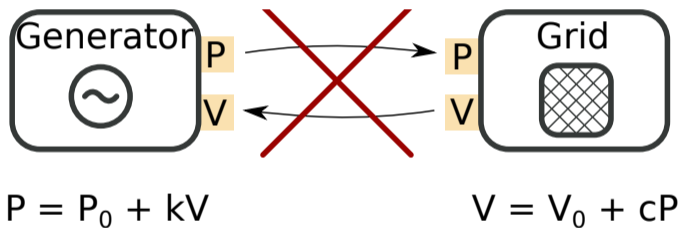


$$V = V_0 + cP$$



Mosaik Quirk # 2: No algebraic closure

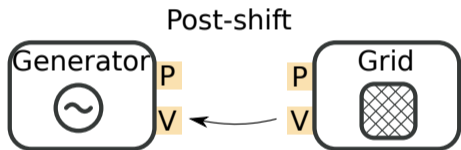
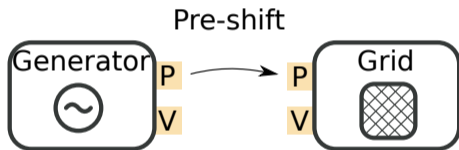
Droop-controlled generator and weak link grid:



Mosaik Quirk # 2: No algebraic closure

Droop-controlled generator and weak link grid:

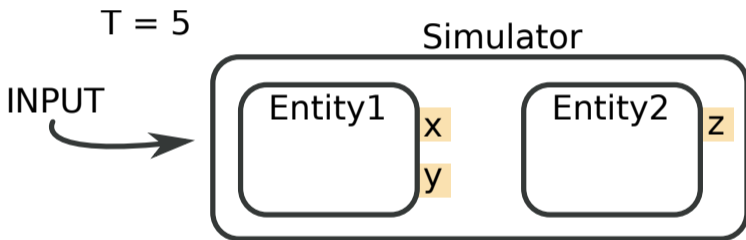
t=20



Use shifted links to break algebraic closures.

Mosaik Quirk # 3: Inputs: Simulators and entities

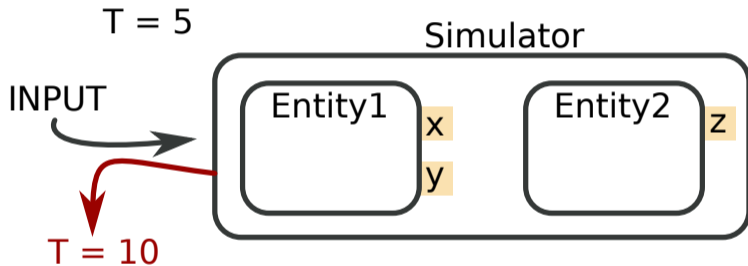
Inputs are handled by simulators, not entities.



INPUT may contain information for Entity1, Entity2, or both - simulators handle exchanging information.

Mosaik Quirk # 4: Lazy evaluation

When a simulator is given inputs, it must return information to Mosaik telling it how long the values are valid for.



Mosaik orchestrator guarantees that the simulator will not be called again until this time.

Allows adaptive behaviour, BUT no way to "wake up" some simulator depending on an event.

The HEMS system in Mosaik

