Under the hood: Orca framework and extensions

Sam Maurer

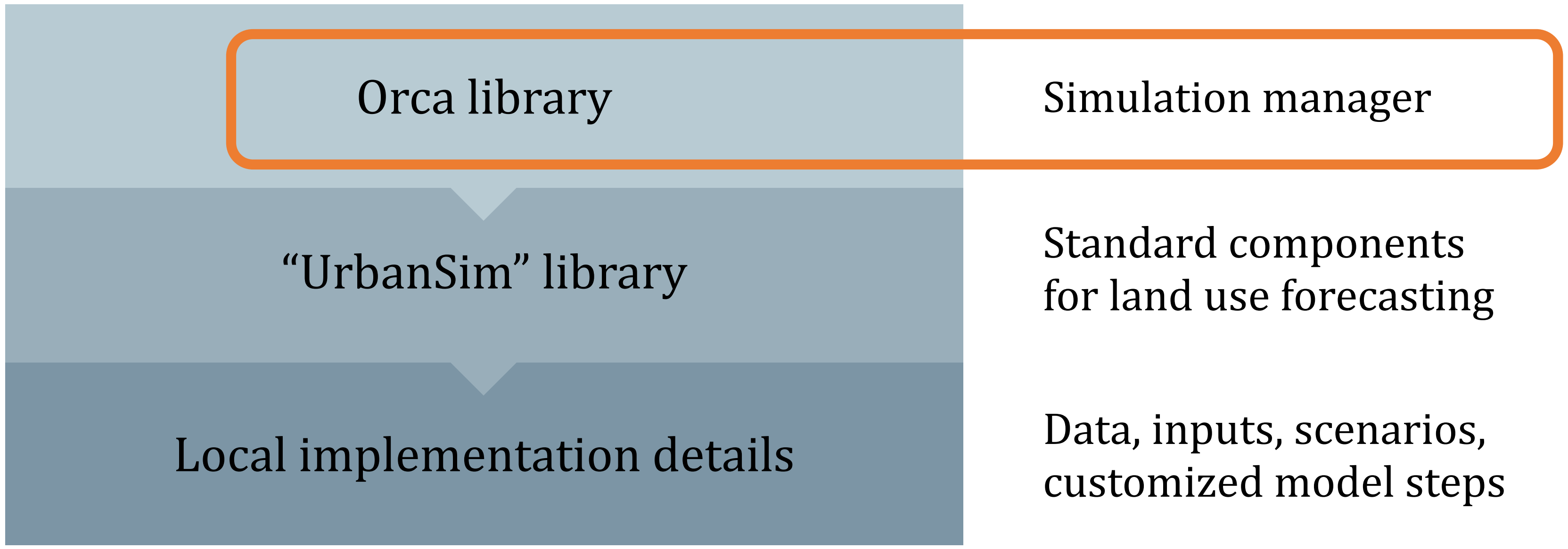
U.C. Berkeley & UrbanSim Inc.

UrbanSim User Meeting

November 4, 2016

What’s Orca?

Architecture of UrbanSim running locally


Orca library
Simulation manager
“UrbanSim” library
Standard components for land use forecasting
Local implementation details
Data, inputs, scenarios, customized model steps

This talk: Orca framework and extensions

1. What does Orca do? How does it work?
2. New feature: Validating data requirements
3. Demo!

Read about Orca

- http://udst.github.io/orca

Follow this presentation (slides and code demo)

- http://github.com/smmaurer/orca-demos

What does Orca do?

- When you launch an UrbanSim model, Orca starts up first
  - Registers data tables
  - Registers relationships between tables
  - Registers model steps

- To begin a simulation, you tell Orca which model steps to run

```
orca.run([‘prices’, ‘household_relocation’, ‘housing_development’])
```

- Orca executes the steps, and manages changes to the data tables
  - New households, new developments, changing characteristics

“Pipeline orchestration”

- Coordinating the execution of a sequence of computational tasks
- Other tools for this
  - Airflow (http://pythonhosted.org/airflow)
  - Luigi (https://github.com/spotify/luigi)
- Orca’s specialties
  - Optimized for iterative simulation (i.e., many years in sequence)
  - Optimized for fast network calculations and statistical forecasting

Orca tips and tricks

- **Define virtual data columns**

  http://udst.github.io/orca/core.html#columns

- **Control when data is cached and when it's recalculated**

  http://udst.github.io/orca/core.html#caching

- **Merge tables automatically**

  http://udst.github.io/orca/core.html#automated-merges

New feature: Validating data requirements!

- Motivation
  - The most common source of errors in running a simulation is when data doesn’t match your expectations — either because of oversight or because of model complexity
  - How can we better avoid and recover from these problems?
- Solution
  - **New syntax** for describing data requirements (data types, max and min values, missing value coding, primary/foreign key relationships)
  - **Easy workflows** for documenting expected data characteristics at different points within a simulation
  - **Fast tools for testing** whether data meets these expectations

Use cases for Orca data validation

- **Validating input data**
  - Missing values? Outliers? Duplication?
  - Write a spec listing the requirements for the data, and Orca will run a customized battery of hierarchical tests
- **Guardrails around model steps**
  - Complicated scenario dependencies? Unexpected errors?
  - Write specs listing the requirements and output of each model step, and Orca will run dynamic tests throughout the simulation, raising descriptive errors instead of crashing if there are problems

Technical details

- For now, data validation tools are in a separate library: **Orca_test**
  - http://github.com/udst/orca_test
  - No changes to existing Orca API
- Data specs are stored in nested **classes**

```
# Define a specification
o_spec = OrcaSpec('my_spec',

    TableSpec('buildings',
        ColumnSpec('building_id', primary_key=True),
        ColumnSpec('residential_price', min=0, missing=False)),

    TableSpec('households',
        ColumnSpec('building_id', foreign_key='buildings.building_id', missing_val_coding=-1)),

    TableSpec('residential_units', registered=False),

    InjectableSpec('rate', greater_than=0, less_than=1))
```

Technical details, continued

- You validate data by **asserting** an OrcaSpec, and an OrcaAssertionError is raised if it fails
- Spec components have a semantic hierarchy
  - Max/min –> numeric –> can be generated –> is registered
- Designed to minimize computation and memory overhead

Our experiences with Orca data validation

- U.C. Berkeley research fork of Bay Area UrbanSim
    - Model step specs have improved the quality of our code
    - Documentation is easier, fewer errors deploying on new machines, greater confidence in the correctness of output
- UrbanSim Cloud Platform
    - Used for validation of uploaded data
    - Incorporated into quality control workflow for auto-generated models

Learn more

Orca

- http://udst.github.io/orca
- http://github.com/udst/orca

Orca data validation tools

- http://github.com/udst/orca_test

This presentation

- http://github.com/smmaurer/orca-demos

- Sam Maurer: maurer@berkeley.edu