Sirepo: Domain-Rich Computational Physics Gateway*

Robert Nagler, Paul Moeller & David Bruhwiler – RadiaSoft LLC
Maksim Rakitin & Oleg Chubar – Brookhaven National Labs
nagler@radiasoft.net – rsl.link/sgci18

radiasoft
radiasoft.net

sirepo
sirepo.com

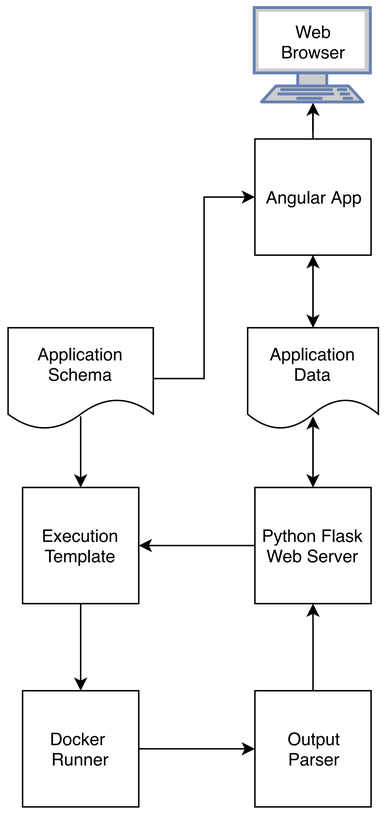
Fork me on GitHub

Design Objectives

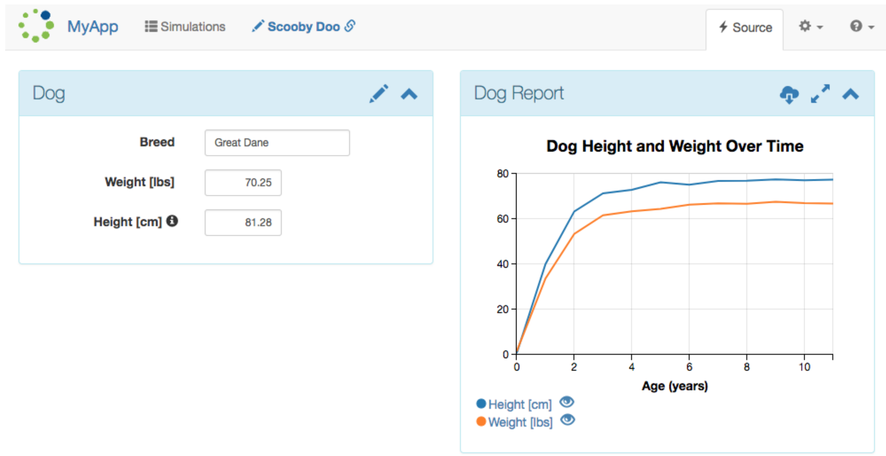
- Provide actual novice to expert workflows in GUI
- Integrate native computational codes, unmodified
- Support real users with distributed computation
- Canonical input/output to simplify implementation
- Multi-modal sharing between users
- Import/export between native and canonical formats
- Escape to CLI for experts and alternative execution modes

Architecture

- Sirepo is a collection of “apps” declared in schemas
- Angular GUI uses schema to derive views and routes
- App data is stored as JSON in a file and validated against the schema
- Execution engine applies app data to a template
- Post-processing occurs on output *in situ* to produce reports
- Views are laid out in HTML templates for local routes (pages)



Demo App



UI Template

```
<div class="container-fluid">
  <div class="row">
    <div class="col-md-6">
      <div data-basic-editor-panel="" data-view-name="dog"></div>
    </div>
    <div class="col-md-6">
      <div data-report-panel="parameter" data-model-name="dogReport"></div>
    </div>
  </div>
</div>
```

Application Data

```
{
  "models": {
    "dog": {
      "breed": "Great Dane",
      "gender": "male",
      "height": 81.28,
      "weight": 70.25
    },
    "dogReport": {},
    "simulation": {
      "name": "Scooby Doo"
    }
  },
}
```

Execution Template

```
#!/usr/bin/env python
import subprocess
params = '''---
breed: "{{ dog.breed }}"
gender: "{{ dog.gender }}"
height: {{ dog.height }}
weight: {{ dog.weight }}
'''
with open('{{ input_name }}', 'w') as f:
    f.write(params)
subprocess.check_call(
    ['hundli', '{{ input_name }}', '{{ output_name }}'],
)
```

Application Schema

```
{
  "localRoutes": {
    "source": {
      "config": {
        "controller":
          "MyAppSourceController as source",
        "templateUrl":
          "/static/html/myapp_source.html"
      }
    }
  },
  "enum": {
    "Gender": [
      ["male", "Male"],
      ["female", "Female"]
    ]
  },
  "model": {
    "dog": {
      "breed": ["Breed", "String"],
      "gender": ["Gender", "Gender"],
      "height": [
        "Height [cm]", "Float", null,
        "Distance from front paws to withers"
      ],
      "weight": ["Weight [lbs]", "Float"]
    },
    "dogReport": {}
  },
  "view": {
    "dog": {
      "title": "Dog",
      "basic": [
        "breed",
        "weight",
        "height"
      ],
      "advanced": [
        "breed",
        "gender",
        "weight",
        "height"
      ]
    },
    "dogReport": {
      "title": "Dog Report",
      "advanced": []
    }
  }
}
```

Future Work

- Output parsing could be better automated.
- Investigate rule-based execution template generation.
- Support supercomputer and traditional “sbatch” style execution.
- Add admin interface to job runner.
- Refactoring Javascript for better sharing between apps.

rsl.link/sgci18

This material is based upon work supported by the U.S. Department of Energy, Office of Science under Award Nos. DE-SC0011237 and DE-SC0015209 from the Office of Basic Energy Sciences, Award No. DE-SC0011340 from the Office of High Energy Physics, Award No. DE-SC0015212 from the Office of Nuclear Physics and Award No. DE-SC0017162 from the Office of Advanced Scientific Computing Research.

U.S. DEPARTMENT OF ENERGY
Office of Science