Modelica Parser Documentation

Release 0.2.1

Dongping XIE

Jan 17, 2019

Contents

Contents:

CHAPTER 1

modparc

modparc is a Modelica parser in Python based on parser combinator.

- Free software: GNU General Public License v3
- Source code: https://github.com/xie-dongping/modparc.
- Documentation: https://modparc.readthedocs.io.

Contents

1.1 Quickstart

Install the package from PyPI:

```
$ pip install modparc
```

To parse a Modelica source file *"your_modelica_file.mo"*:

```
import modparc
model_definition = modparc.parse_file("your_modelica_file.mo")
```

To list all the equations in the *model_definition* instance:

```
all_equations = model_definition.search('Equation')
for equation in all_equations:
    print(equation.code())  # The code of the equation as string
```

To get the name of the model loaded:

```
print(model_definition.name())  # get the name of the stored class
print(model_definition.class_type())  # get the type of the class
```

1.2 Features

- Experimentally parses Modelica Standard Library 3.2.1
- Search element of a certain class

1.3 Known Issues

- Handling tokenization of Q-IDENT and comments, which comes first?
- Assertion syntax not defined in Modelica specification
- Default recursion depth is not enough for long vector literals
- Cyclic import is neccessary for the Modelica syntax definition

1.4 Credits

This package was created with Cookiecutter and the audreyr/cookiecutter-pypackage project template.

The test cases used code from the **'ModelicaByExample library (MIT License by Michael Tiller)'_**.

CHAPTER 2

Installation

2.1 Stable release

To install Modelica Parser, run this command in your terminal:

```
$ pip install modparc
```

This is the preferred method to install Modelica Parser, as it will always install the most recent stable release.

If you don't have pip installed, this Python installation guide can guide you through the process.

2.2 From sources

The sources for Modelica Parser can be downloaded from the Github repo.

You can either clone the public repository:

```
$ git clone git://github.com/xie-dongping/modparc
```

Or download the tarball:

```
$ curl  -OL https://github.com/xie-dongping/modparc/tarball/master
```

Once you have a copy of the source, you can install it with:

```
$ python setup.py install
```

CHAPTER 3

Usage

To use modparc in a project:

```
import modparc
with open("your_modelica_file.mo", 'r') as f:
    modelica_source_code = f.read()
    model_definition = modparc.parse(modelica_source_code)
```

To list all the equations in the *model_definition* instance:

```
all_equations = model_definition.search('Equation')
for equation in all_equations:
    print(equation.code())  # The code of the equation as string
```

One could also parse a certain syntax element in Modelica:

```
import modparc
from modparc.syntax import tokenize
source_code = """
              if init==InitializationOptions.FixedPopulation then
                population = initial_population;
              elseif init==InitializationOptions.SteadyState then
                der(population) = 0;
              else
              end if
              """
tokens_list = tokenize(source_code)
if_equation_element = modparc.syntax.equations.if_equation(tokens_list)
sub_equations = if_equation_element.search('Equation')
for equation in sub_equations:
    print(equation.code())  # The code of the equation as string
```

CHAPTER 4

Contributing

Contributions are welcome, and they are greatly appreciated! Every little bit helps, and credit will always be given.

You can contribute in many ways:

4.1 Types of Contributions

4.1.1 Report Bugs

Report bugs at https://github.com/xie-dongping/modparc/issues.

If you are reporting a bug, please include:

- Your Python version and the result of a *pip list*
- Detailed steps to reproduce the bug, ideally a test case.

4.1.2 Fix Bugs

Look through the GitHub issues for bugs. Anything tagged with "bug" and "help wanted" is open to whoever wants to implement it.

4.1.3 Implement Features

Look through the GitHub issues for features. Anything tagged with "enhancement" and "help wanted" is open to whoever wants to implement it.

4.1.4 Write Documentation

Modelica Parser could always use more documentation, whether as part of the official Parser docs, in docstrings, or even on the web in blog posts, articles, and such.

4.1.5 Submit Feedback

The best way to send feedback is to file an issue at https://github.com/xie-dongping/modparc/issues.

If you are proposing a feature:

- Explain in detail how it would work.
- Keep the scope as narrow as possible, to make it easier to implement.
- Remember that this is a volunteer-driven project, and that contributions are welcome :)

4.2 Get Started!

Ready to contribute? Here's how to set up *modparc* for local development.

1. Fork the *modparc* repo on GitHub.
2. Clone your fork locally:

```
$ git clone git@github.com:your_name_here/modparc.git
```

3. Install your local copy into a virtualenv. Assuming you have virtualenvwrapper installed, this is how you set up your fork for local development:

```
$ mkvirtualenv modparc
$ cd modparc/
$ python setup.py develop
```

4. Create a branch for local development:

```
$ git checkout -b name-of-your-bugfix-or-feature
```

   Now you can make your changes locally.

5. When you're done making changes, check that your changes pass flake8 and the tests, including testing other Python versions with tox:

```
$ flake8 modparc tests
$ python setup.py test or py.test
$ tox
```

   To get flake8 and tox, just pip install them into your virtualenv.

6. Commit your changes and push your branch to GitHub:

```
$ git add .
$ git commit -m "Your detailed description of your changes."
$ git push origin name-of-your-bugfix-or-feature
```

7. Submit a pull request through the GitHub website.

4.3 Pull Request Guidelines

Before you submit a pull request, check that it meets these guidelines:

1. The pull request should include tests, unless it is a minor fix.

2. If the pull request adds functionality, the docs should be updated. Put your new functionality into a function with a docstring, and add the feature to the list in README.rst.
3. The pull request should work for Python 2.7, 3.3, 3.4 and 3.5, and for PyPy. Check https://travis-ci.org/xie-dongping/modparc/pull_requests and make sure that the tests pass for all supported Python versions.

4.4 Tips

To run a subset of tests:

```
$ py.test tests.test_modparc
```

CHAPTER 5

Credits

5.1 Author

- Dongping XIE <dongping.xie.tud@gmail.com>

5.2 Contributors

None yet. Why not be the first?

CHAPTER 6

History

6.1 0.1.5 (2016-10-22)

- First release on PyPI.

6.2 0.2.0 (2016-10-22)

- Get names and types of the defintions
- Roundtripping of the defintions

CHAPTER 7

Indices and tables

- genindex
- modindex
- search