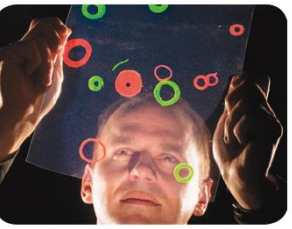
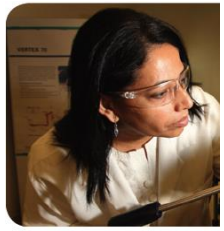




Sandia National Laboratories

Exceptional service in the national interest

1. Overview of Pyomo

PYOMO

U.S. DEPARTMENT OF ENERGY

NNSA National Nuclear Security Administration

CCR Center for Computing Research

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000. SAND NO. 2011-XXXXP

Pyomo Overview

Sandia National Laboratories

Idea: a Pythonic framework for formulating optimization models

- Provide a natural syntax to describe mathematical models
- Formulate large models with a concise syntax
- Separate modeling and data declarations
- Enable data import and export in commonly used formats

Highlights:

- Python provides a clean, intuitive syntax
- Python scripts provide a flexible context for exploring the structure of Pyomo models

```
# simple.py
from pyomo.environ import *

M = ConcreteModel()
M.x1 = Var()
M.x2 = Var(bounds=(-1,1))
M.x3 = Var(bounds=(1,2))
M.o  = Objective(
         expr=M.x1**2 + (M.x2*M.x3)**4 + \
              M.x1*M.x3 + \
              M.x2*sin(M.x1+M.x3) + M.x2)

model = M
```

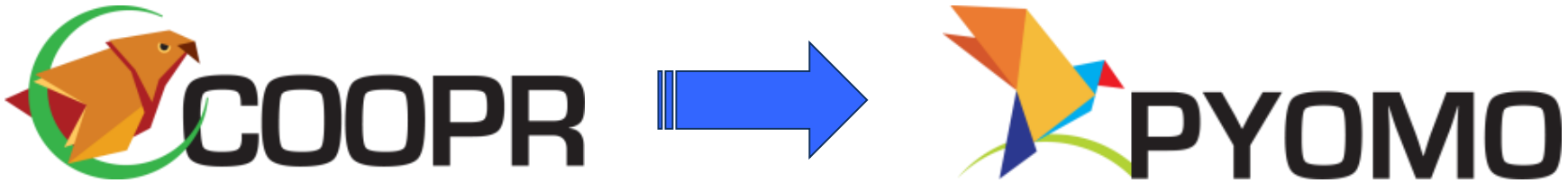
CCR Center for Computing Research

PYOMO COIN-OR

Overview

- What happened to Coopr?
- Three really good questions:
  - Why another Algebraic Modeling Language (AML)?
  - Why Python?
  - Why open-source?
- Pyomo: Software library infrastructure
- Pyomo: Team overview and collaborators / users
- Where to find more information…

What Happened to Coopr?



- Users were installing Coopr but using Pyomo
  - Pyomo modeling extensions were not distinct enough
  - Researchers cited “Coopr/Pyomo”
- Users/Developers were confused by the `coopr` and `pyomo` commands
- Developers were coding in Coopr but talking about Pyomo

We needed to provide clear branding this project!

Optimization Modeling

Sandia National Laboratories

Goal:

- Provide a natural syntax to describe mathematical models
- Formulate large models with a concise syntax
- Separate modeling and data declarations
- Enable data import and export in commonly used formats

Impact:

- Robustly model large constraint matrices (e.g. for MILPs)
- Integrated support of automatic differentiation for complex nonlinear models

Examples:

- AMPL, GAMS, AIMMS, …
- OptimJ, FlopCPP, PuLP, JuMP, …

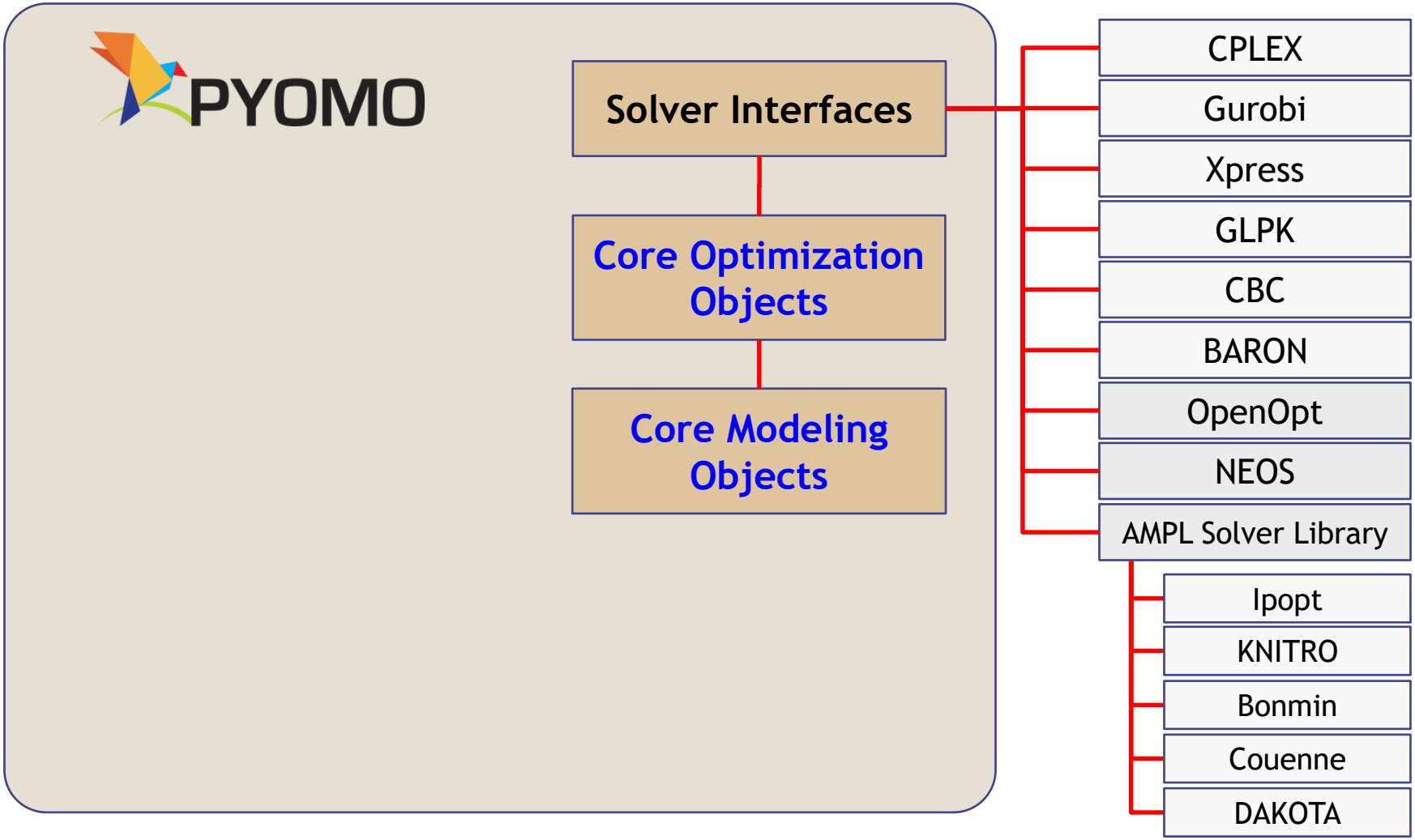
Why Model in Python?

- Full-Featured Library
  - Language features includes functions, classes, looping, namespaces, etc
  - Introspection facilitates the development of generic algorithms
  - Python's clean syntax facilitates rapid prototyping
- Open Source License
  - No licensing issues w.r.t. the language itself
- Extensibility and Robustness
  - Highly stable and well-supported
- Support and Documentation
  - Extensive online documentation and several excellent books
  - Long-term support for the language is not a factor
- Standard Library
  - Includes a large number of useful modules
- Portability
  - Widely available on many platforms

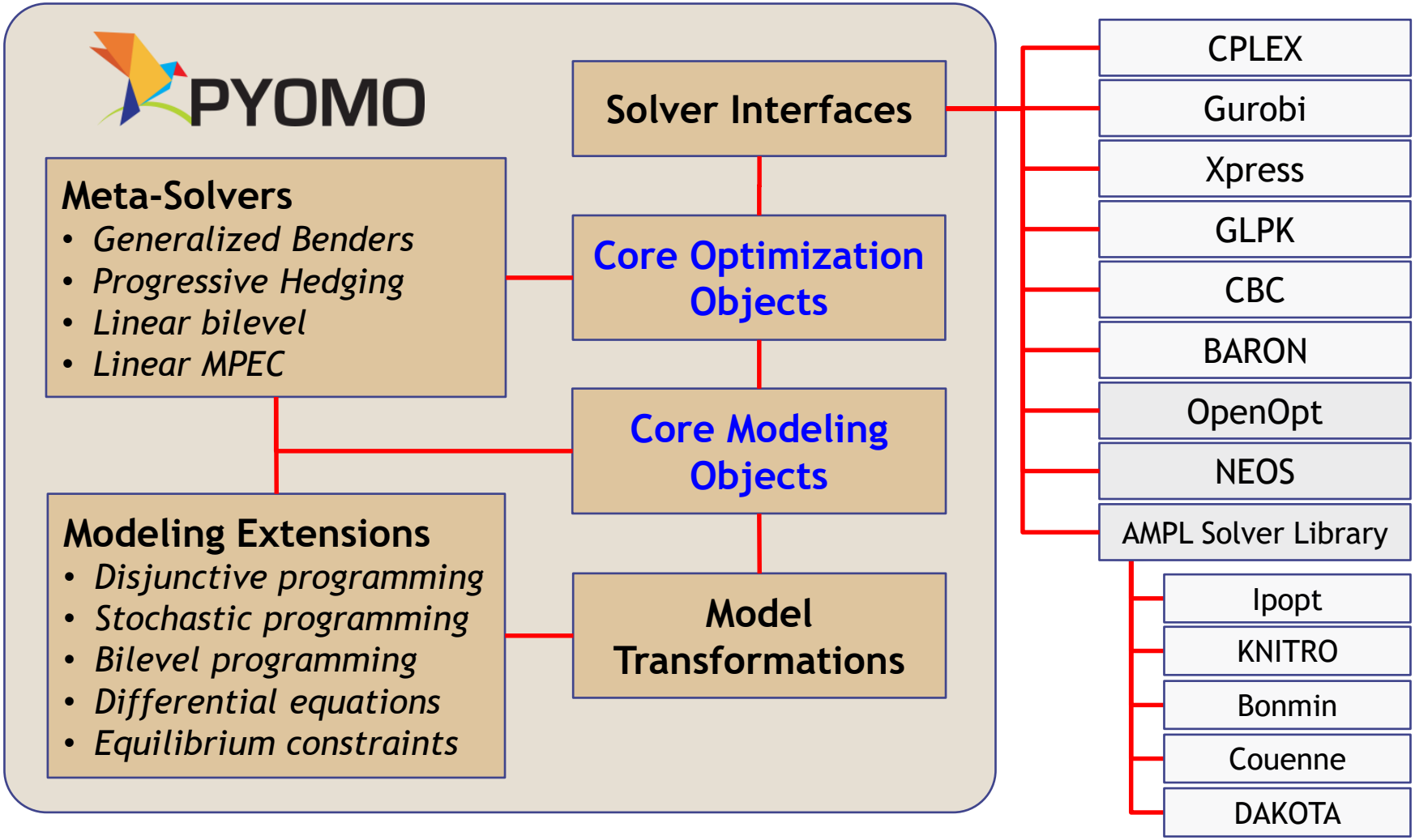
Why Open Source?

- Transparency and reliability
- Foster community involvement
  - Extend the modeling language
  - Develop new solvers / algorithms
  - Interface with additional external utilities
  - “Stone Soup” model
- Flexible licensing
  - Pyomo released under 3-clause BSD license
  - No restrictions on deployment or commercial use

Pyomo at a Glance



Pyomo at a Glance



Survey of Python Modeling Tools

- Pyomo
  - Supports concrete/abstract modeling for LP/MILP/NLP models
  - Modeling extensions for stochastic programming, bilevel, MPEC, etc
  - Separate model objects
- PuLP
  - Supports concrete modeling for LP/MILP models
  - Separate model objects
  - Simple object model
- APLEpy
  - Supports concrete modeling for LP/MILP models
  - Single global model object
- PyMathProg, pyglpk, cplex, gurobi
  - Python interfaces for specific solver tools

More than *just* mathematical modeling

Sandia National Laboratories

Scripting

- Construct models using native Python data
- Iterative analysis of models leveraging Python functionality
- Data analysis and visualization of optimization results

Model transformations (a.k.a. reformulations)

- Automate generation of one model from another
- Leverage Pyomo's object model to apply transformations sequentially
- E.g.: relax integrality, GDP -> Big M

Meta-solvers

- Integrate scripting and/or transformations into optimization solver
- Leverage Python's introspective nature to build "generic" capabilities
- E.g.: progressive hedging, SP extensive form -> MIP

CCR Center for Computing Research

PYOMO COIN-OR

Who Uses Pyomo?

- Students
  - Rose-Hulman, UC Davis, U Texas, Iowa State, NPS
- Researchers
  - Government laboratories
    - Sandia National Labs, Lawrence Livermore National Lab, Los Alamos National Lab, National Energy Technology Lab, Federal Energy Regulation Commission
  - Universities
    - UC Davis, TAMU, Rose-Hulman, UT, USC, GMU, Iowa State, NCSU, U Washington, NPS, U de Santiago de Chile, U Pisa, …
  - Companies

Who Uses Pyomo?

- Software Projects
  - TEMOA – Energy economy optimization models
  - Minpower – Power systems toolkit
  - Water Security Toolkit – Planning/Response for water contamination
  - SolverStudio – Excel plugin for optimization modeling

For More Information

See the Pyomo homepage

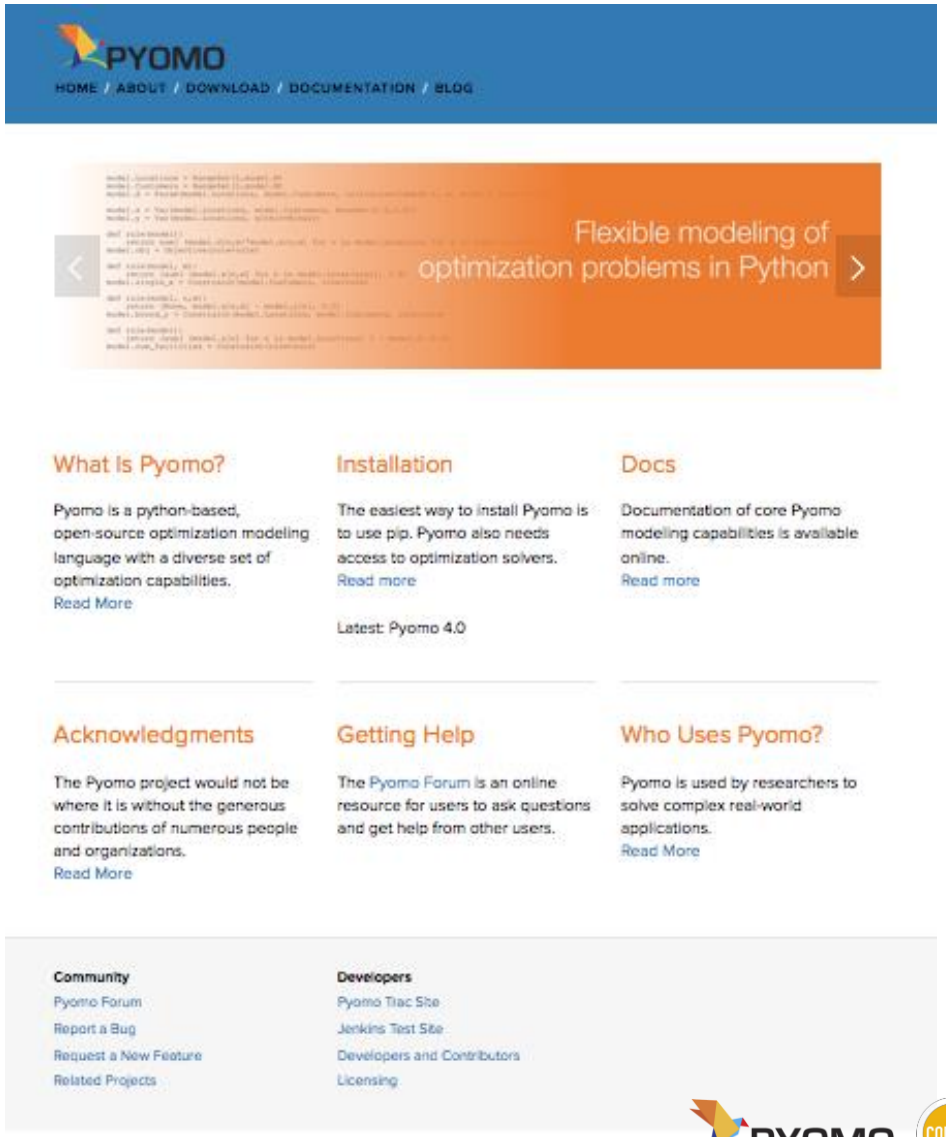
- www.pyomo.org

The Pyomo homepage provides a portal for:

- Online documentation
- Installation instructions
- Help information
- Developer links

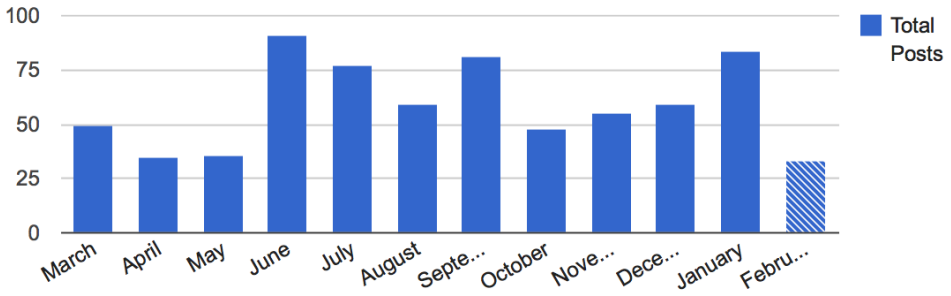
Coming soon:

- A gallery of simple examples

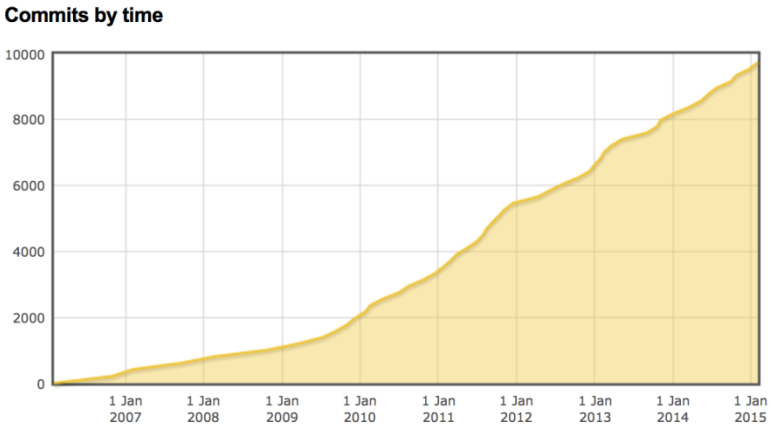


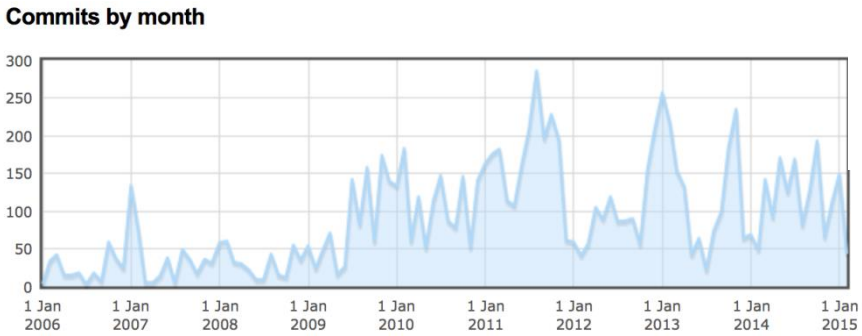
Development, Community Activity

- Pyomo Forum
  - Active discussion list



- Active developer community





Acknowledgements

- Sandia National Laboratories
  - William Hart
  - Jean-Paul Watson
  - John Siirola
  - Francisco Munoz
- University of California, Davis
  - Prof. David L. Woodruff
  - Prof. Roger Wets
- Purdue University
  - Prof. Carl D. Laird
- Oregon State University
  - Gabe Hackebeil
- Carnegie Mellon University
  - Bethany Nicholson