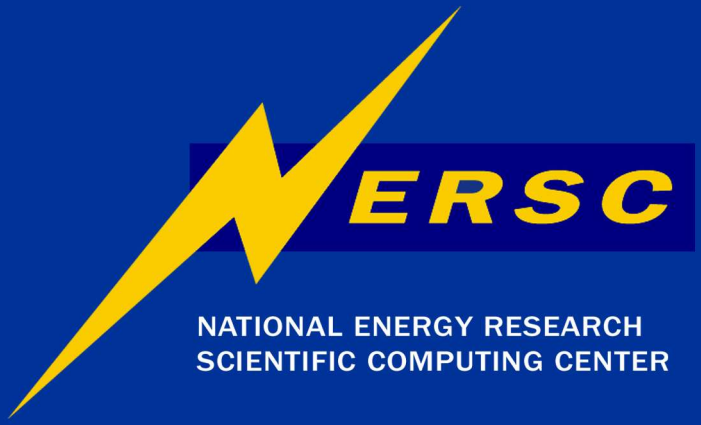




# National Energy Research Scientific Computing Center (NERSC)

## Eclipse as Physicist Work Environment

Wim T.L.P. Lavrijsen, Sebastien Binet
NERSC HENPC, LBNL
CHEP – Mumbai, February 2006

# What is Eclipse?

- ***Eclipse* is:**

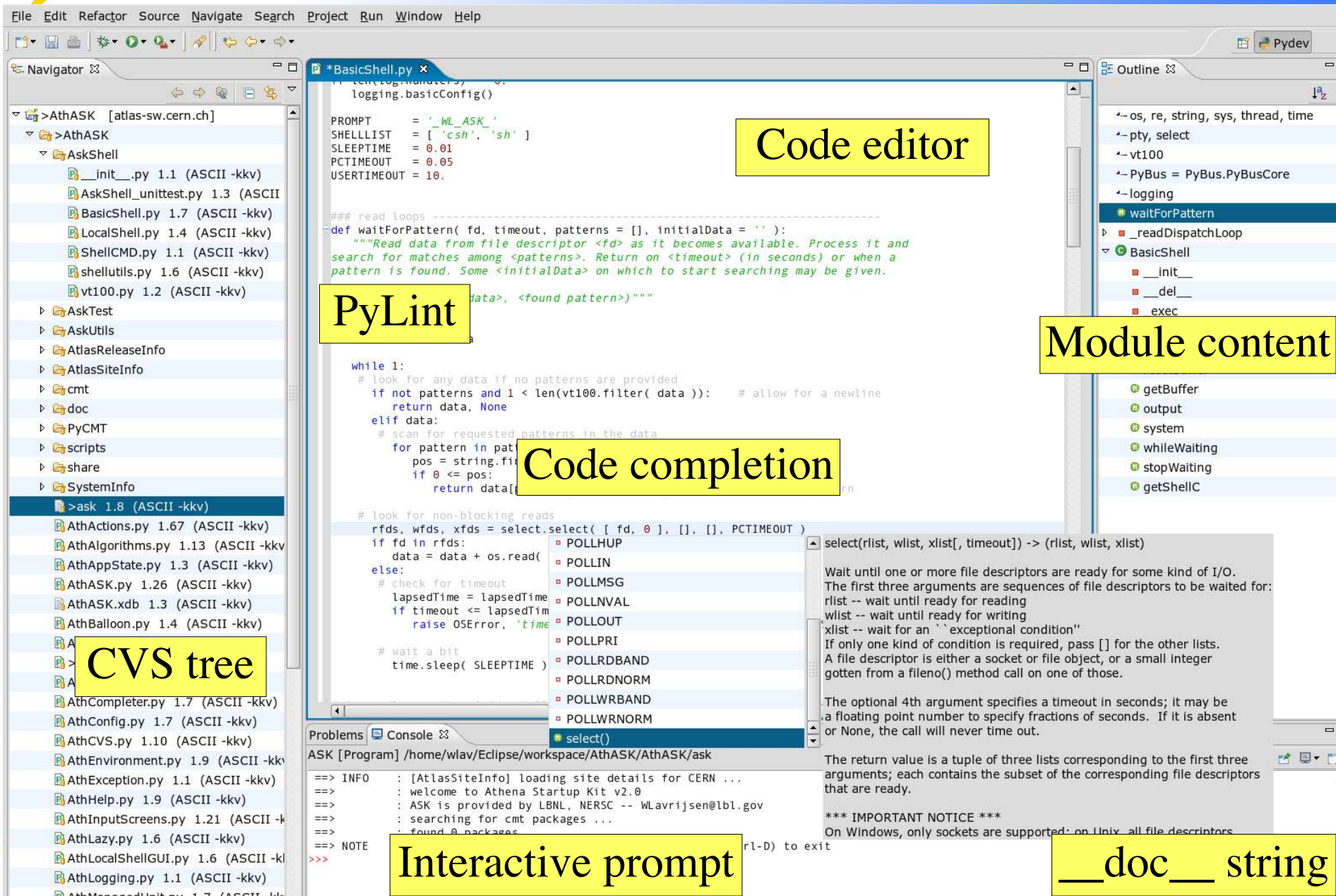
  **an open source community whose projects are focused on providing a vendor-neutral open development platform and application frameworks for building software.**

- **The *Eclipse Foundation* is:**

  **a not-for-profit corporation formed to advance the creation, evolution, promotion, and support of the Eclipse Platform and to cultivate both an open source community and an ecosystem of complementary products, capabilities, and services.**

- **Variety (!) of big-name players:**
  - **IBM, Intel, Borland, CA, BEA, Sybase, Nokia, Wind River, HP, SAP**

# Best known face of Eclipse: IDE



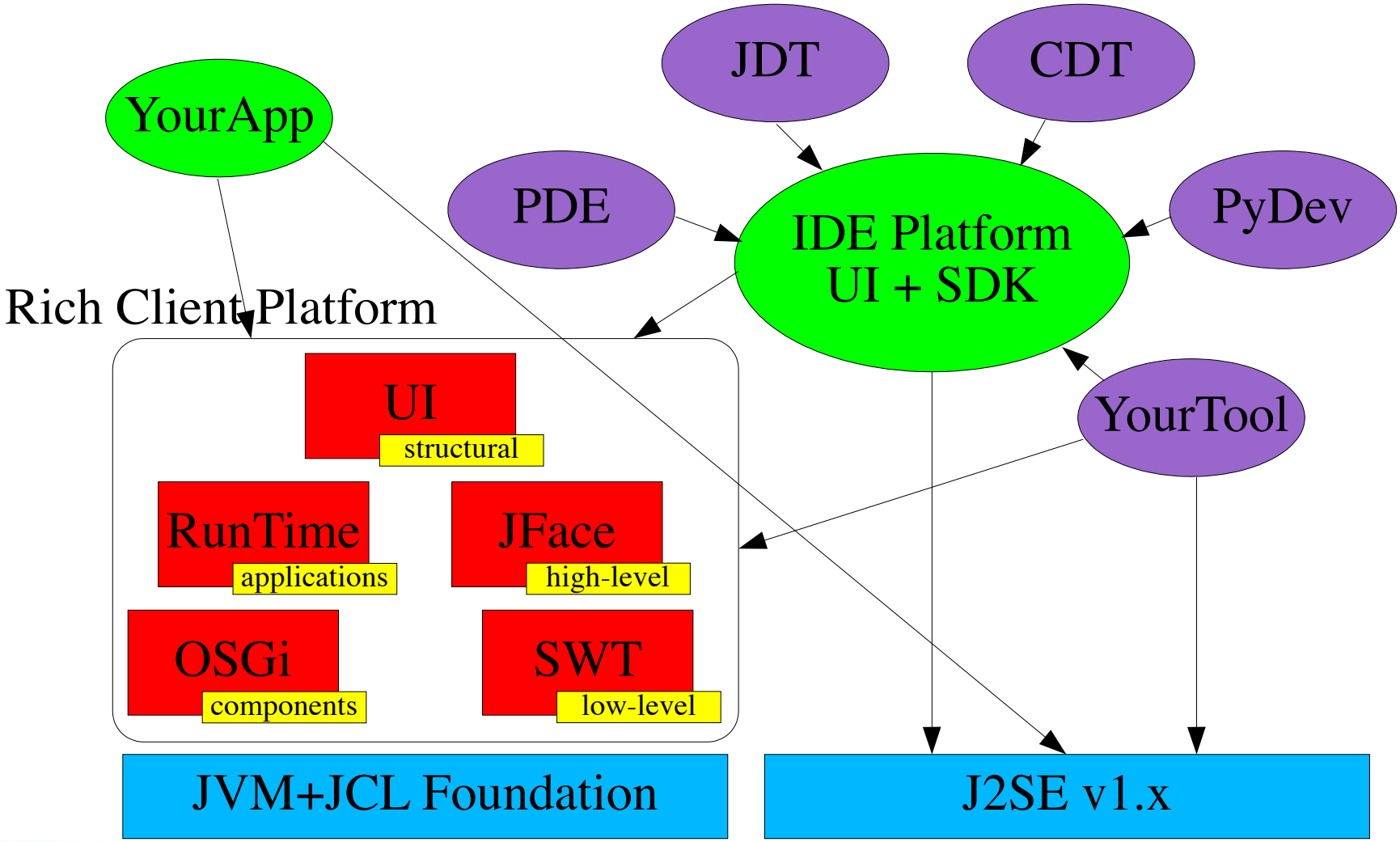
# Looks good ... usable?

- **Need recent hardware (OS/JVM dep.)**
  - Ongoing dev. to improve performance
  - Eclipse itself is nicely responsive
    - Editors, wizards, code-completion, etc.
  - Code compilation slower than shell
  - Memory hungry (as is Atlas Software)
    - E.g. can't index/build ROOT on a 4GB box
- **Redundancy: yet another Java ...**
  - GUI toolkit (SWT)
  - Component model (Equinox)

# Atlas problems to solve

- **Software life cycle**
  - Not part of average physicist's toolkit
  - Neither intuitive, nor explorable
- **Tool integration**
  - Multi-language environment
  - Individual tool interactions
  - Setup/environment difficulties
- **Beginners sandbox**
  - More sophisticated “undo” than “rm -rf”

# IDE is just one application ...




Office of Science
U.S. DEPARTMENT OF ENERGY

# Middleware

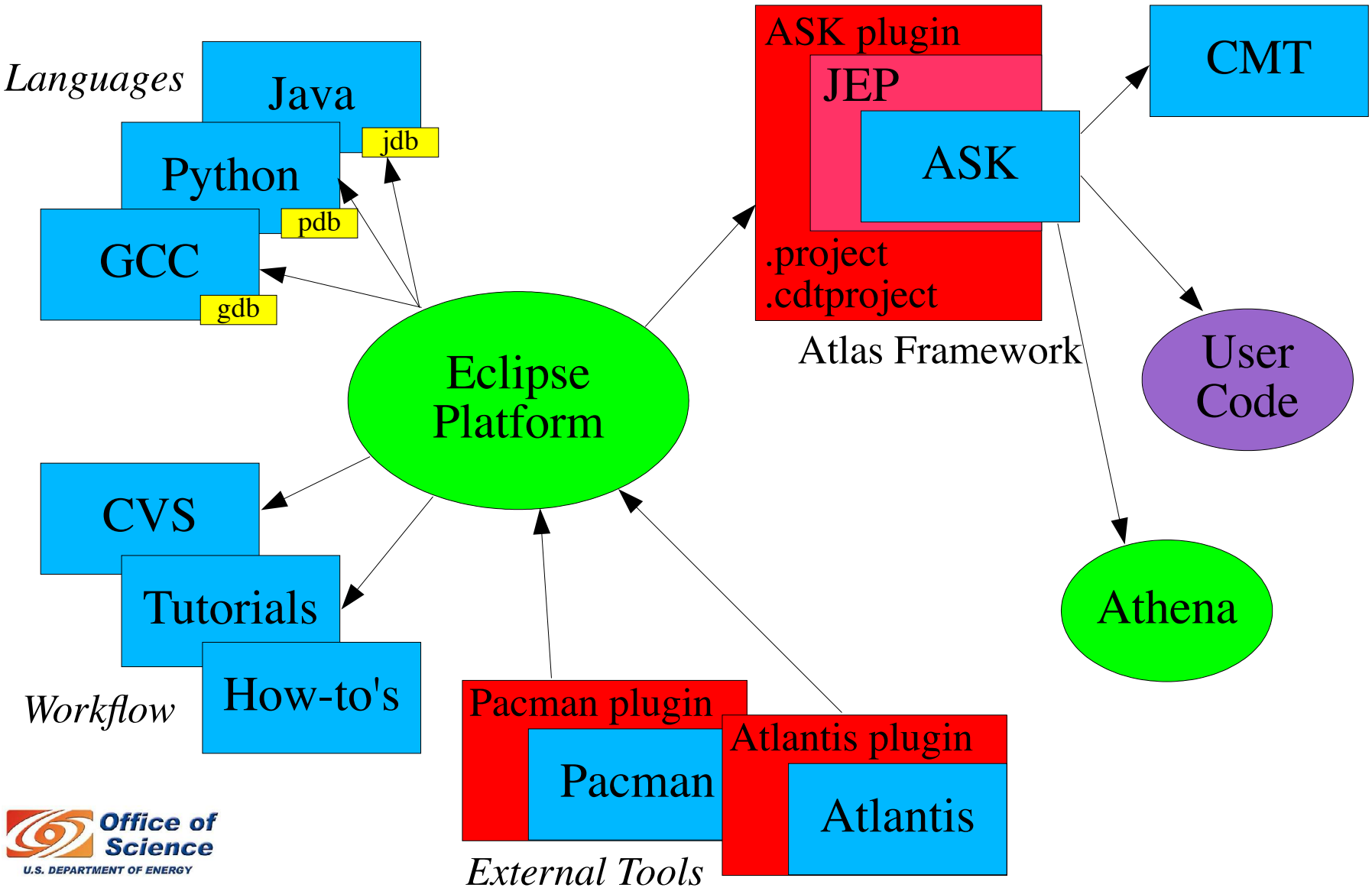
- **Not a GUI, but an *application platform***
  - Tools integration
    - Multi-language, -platform, and -vendor
    - Adaptation, distribution (install/updates)
- **Component model implements OSGi**
  - http://osgi.org (version 4)
  - Plugins/Bundles w/ life- and runtime mgmt
    - Only connect through extension points
    - Each their own “execution space” (i.e. class loader and process environment)

# Open Services Gateway initiative Component Model

- **Complexity solvable by modular sw**
  - *Only* possible if dependencies managable
    - 100% automatic <=> 100% strict (no backdoors)
    - Limited human input ok, if always verified
  - Shared libs, .class files != modular
    - More like a grab bag: access is not controlled
- **Lifetime control starts at installation**
  - Eclipse: from Eclipse startup
- **Connection control through broker**
  - Not (yet) implemented in Eclipse

Office of Science
U.S. DEPARTMENT OF ENERGY

# Tools integration




Office of Science
U.S. DEPARTMENT OF ENERGY

# Athena Startup Kit (ASK)

- **User-space set of tools**
  - Fills voids left by other applications
- **Code generation for Atlas Framework**
  - Algorithms, AlgTools, DataObjects
- **RunTime management**
  - Site abstractions (CERN, BNL, local, etc.)
  - CMT requirements, config, setup
  - POOL file catalog maintainance
- **Module, fully scriptable, CLI, GUI**

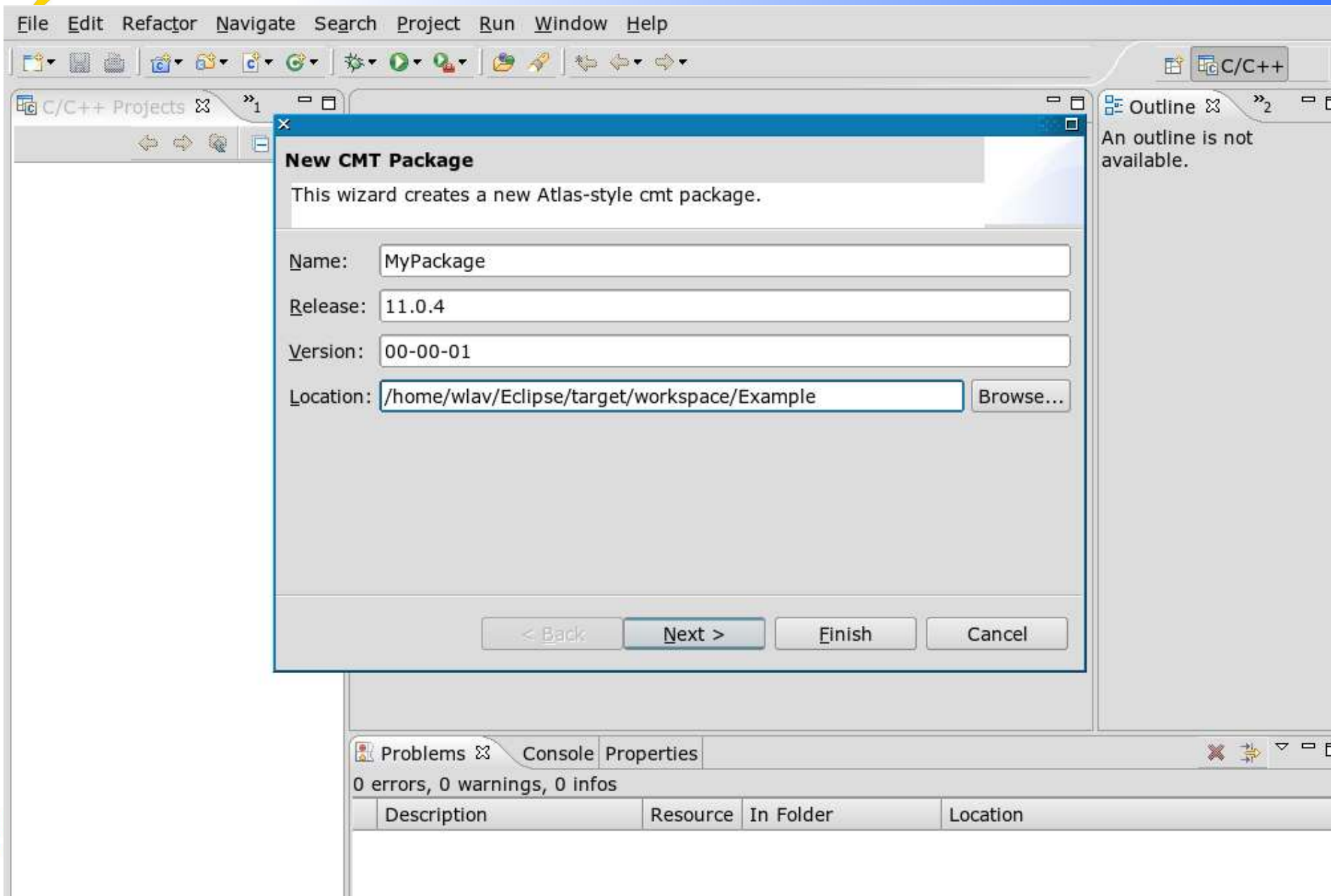
Office of Science
U.S. DEPARTMENT OF ENERGY

# Example Use Case

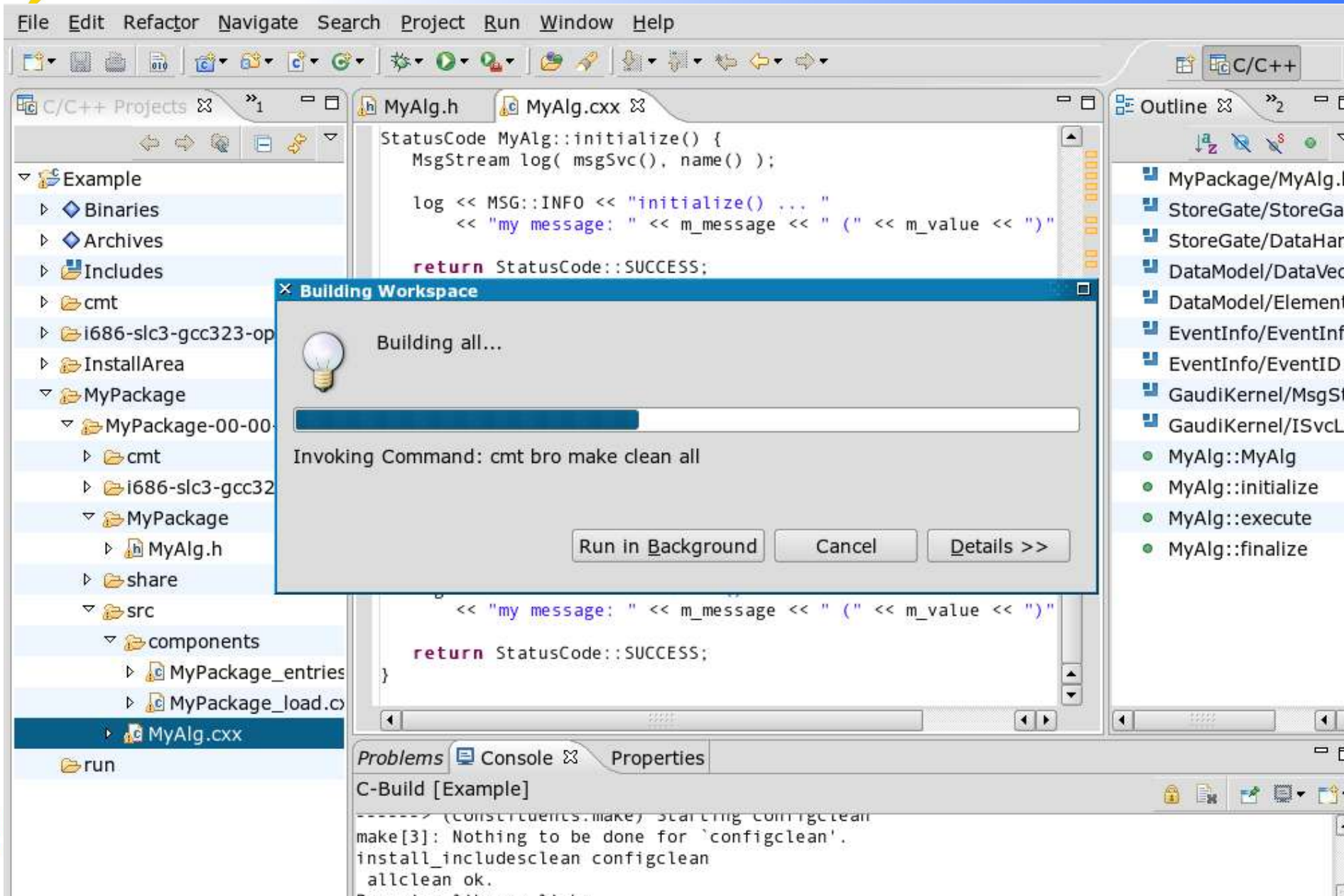
## End-user Workflow

***Code creation, build, run, and check-in***

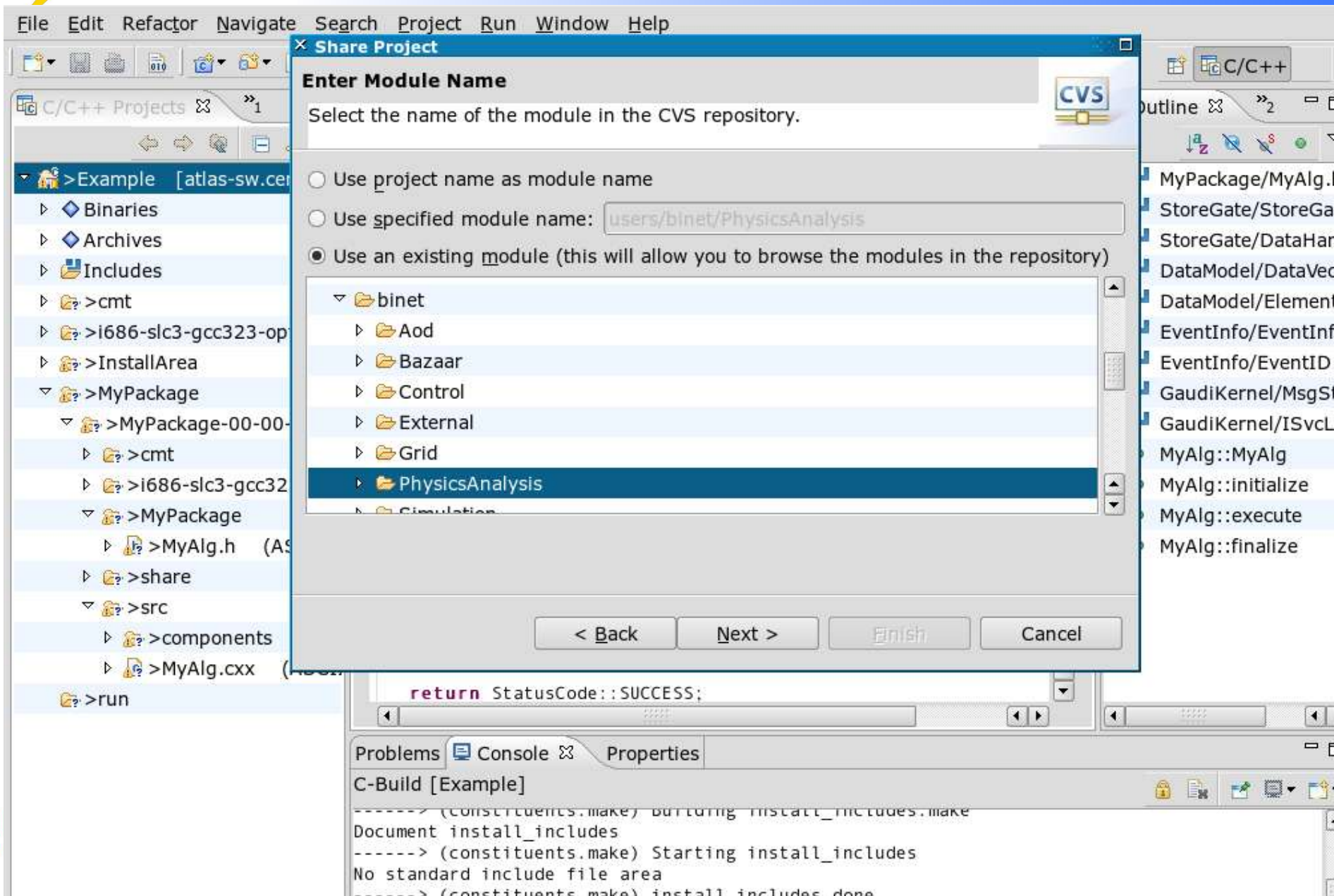
# Wizard for new package



# Build with external tool



# Add to repository



# Conclusions and Outlook

- **Eclipse has lots of potential uses**
  - Impressive, full-featured IDE
  - Individual tool execution environments
  - Allows interactive how-to's
- **Plugin development is easy and clean**
  - Rewarding for post-doc/student
- **Maybe too top-heavy (esp. GUI)**
  - Atlas sw itself has high requirements