

# Runtime Architecture

Stéphane Ducasse

<http://stephane.ducasse.free.fr/> [stephane.ducasse@inria.fr](mailto:stephane.ducasse@inria.fr)

## Execution Model

The Pharo virtual machine (VM) executes compiled code.

- The virtual machine and its plugins are platform specific
- VMs exist for MacOS, Windows, Linux, iOS, ARM, Android

Pharo code is compiled to bytecodes

- Bytecodes are platform neutral instructions

The virtual machine performs dynamically bytecodes to assembly generation

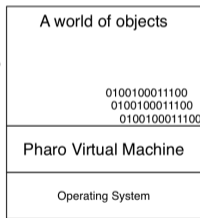
# Virtual Machine

- Pharo.exe is the virtual machine.
- Can be run from command-line or in interactive (UI) mode.
- It executes compiled code.
- Compiled code is packaged in an *image*.
- The virtual machine only needs the *image* to execute programs.

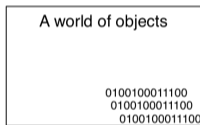
# About Image Files

- *.image* files act as cache of objects
  - ▶ simple objects (points, strings, arrays)
  - ▶ but also compiled classes and compiled methods
- Each time we save the image, all objects are saved to disc.
- At startup we get all the objects we saved
- In particular the PC (program counter) is also saved and restored so frozen execution is restarted at launch time

currently executed image



saved image



# About Change Files

*.changes* file is a tape of all the changes performed in the system.

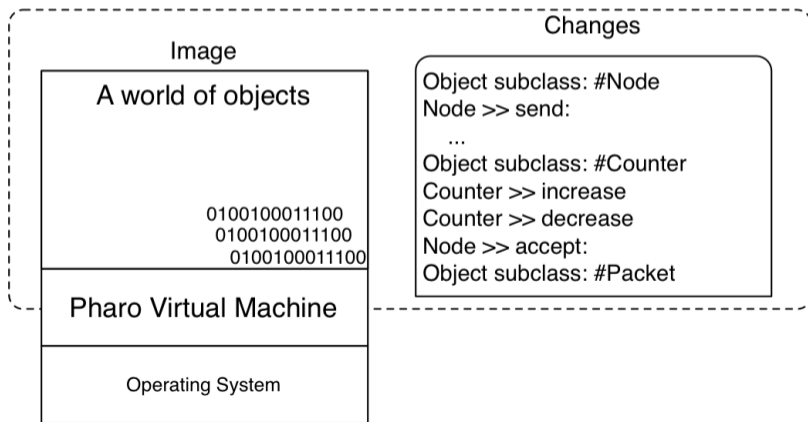
- It logs class creation/deletion, method addition/removal, actions...
- It is used to browse versions.
- It can replay/undo actions.

A change is associated to an image.

- when the browsers want to display class/method definition they look in the change file associate with the current image.

# About Change Files

- A change is associated to an image.
- The image can be executed without the change.
- The change file contains the textual representation of the changes made to the image.



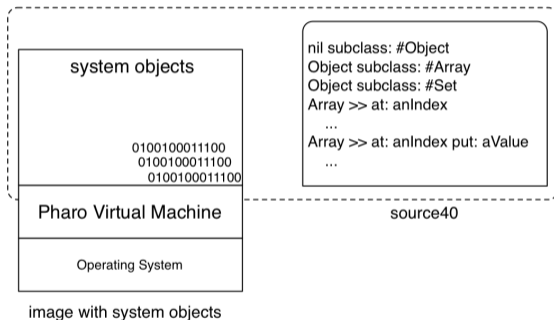
# Save your code using a package and version control system

- Change and image are handy to develop.
- But they are not software engineering artefact
- Always have a loading script that takes an image, load your code, run the test, build your application
- Usually we
  - ▶ save code using a CVS (monticello, git)
  - ▶ use an integration server to build automatically applications.

# About the Source/Changes File

## *PharoXX.sources*

- Contains the textual definition of system classes and predefined objects
- Is read-only.
- Created during release of a new Pharo version.
- Shared by all the users (images)

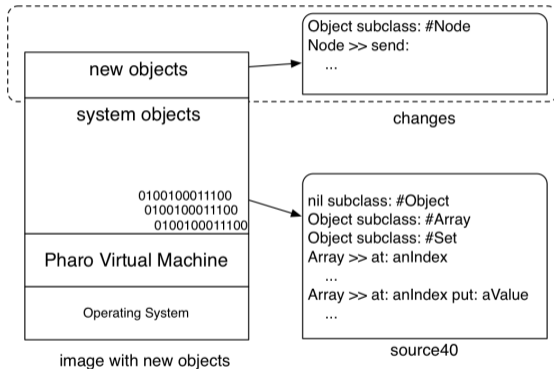




# When you define new classes

During development or code loading,

- custom objects are compiled in the image
- custom definitions are added to the changes file
- Still you can browse the definition of the system class (stored in the *PharoXX.sources*)



# New change management is coming

## Old change management

- Inherited from Smalltalk
- Predates version control systems
- Will change in the future

# Conclusion

- Powerful deployment
- Fast boot-time
- Support micro commits
- Will use modern version control