

Pharo 70

The key challenges

S. Ducasse for the RMoD Team

Points

- Cleaning and stabilising
- 64 bits windows/large images
- Iceberg in production
- Pharo managed on github
- Bootstrap in production
- Retirement of Nautilus -> Calypso + Commander
- New class definition
- Cargo

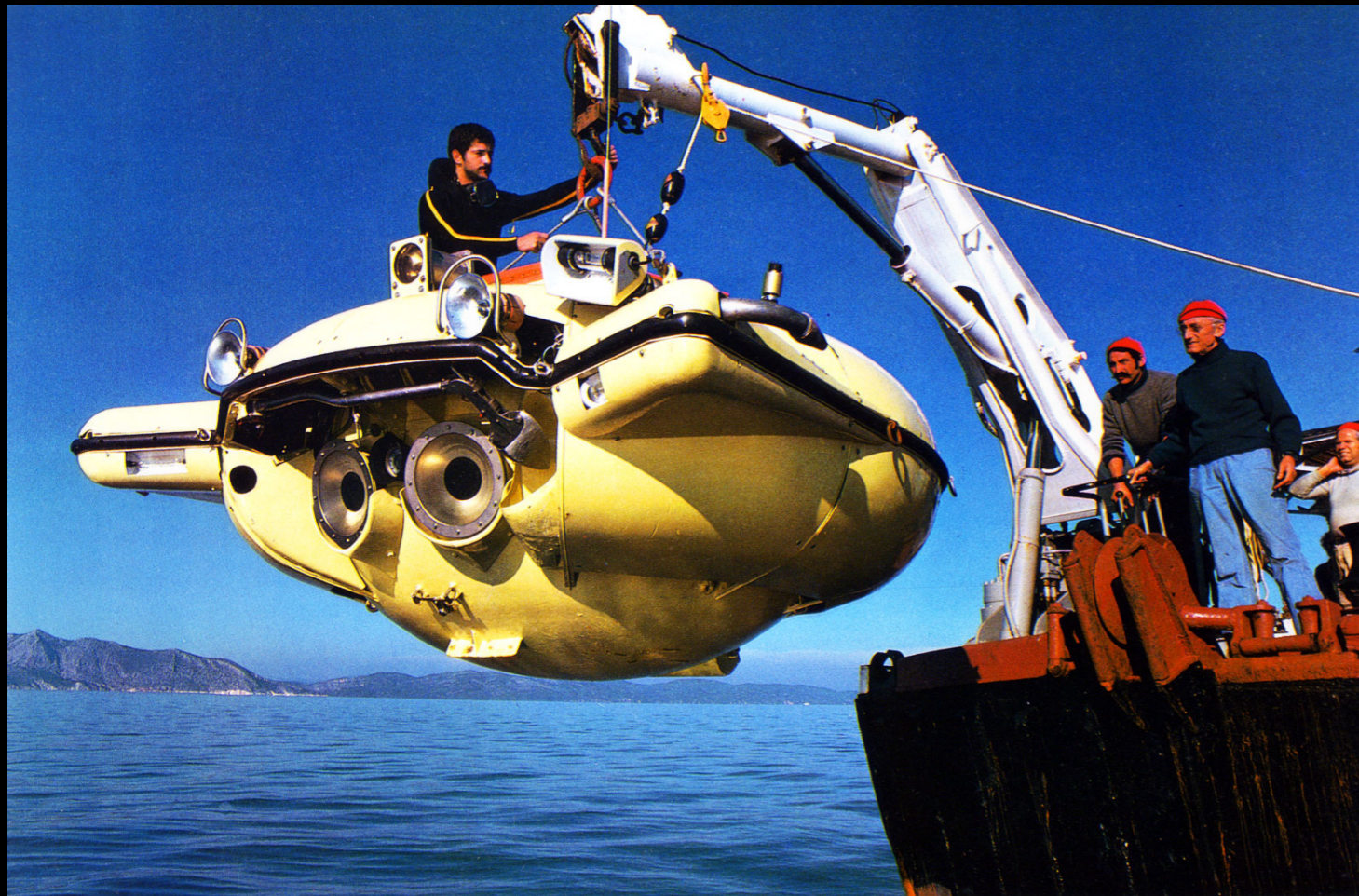
Nautilus



Calypso



Calypso & its submarines



Calypso

- Basis for Remote Browsing
- Fully extensible
- No ifs
- Developed by D. Kudriashov
- Thanks alain for giving us the name

Calypso

- multiple panes
- multiple method definitions
- incomplete method definitions
- scoping...

AbstractConfiguration>>propertyAt:ifAbsent:

- SimpleConfiguration
- BrainstormConfiguration
- Errors
- Junk
- OldPillar
- Test

- AbstractConfiguration
- CCSimpleConfiguration
- CCSimpleConfigurationMock
- SCConfiguration
- CCSimpleSTONConfigurationInterp

- inherited methods
- as yet unclassified
- accessing
- accessing - properties
- copying
- error signalling
- initialization
- logging
- printing
- private
- reflective operations
- subconfigurations
- testing
- overridden

- allProperties
- propertyAt:
- propertyAt:ifAbsent:
- propertyAt:ifAbsentPut:
- propertyAt:ifPresent:ifAbsent:
- propertyAt:put:

Simple Filter...

Packages
 Projects
 | Flat
 Hier.
 | Inst. side
 Class side
 | Methods
 Vars
 | [Class refs.](#)
[Implementors](#)
[Senders](#)

Comment
*AbstractConfigur
propertyAt:ifAbse

```

propertyAt: aKey ifAbsent: aBlock
    "Answer the value of the property ==aKey==, potentially looking in parents, or the result of
    ==aBlock== if the property doesn't exist."

    ^ properties
      at: aKey asSymbol
      ifAbsent: [ self hasParent
                  ifTrue: [ self parent propertyAt: aKey ifAbsent: aBlock ]
                  ifFalse: aBlock ]
    
```

1/9 [1] Format as you read W +L

Updating critiques...

New Message Browser

Implementors of subConfigurationNamed: [2]

Class	Method	Implementor
AbstractConfiguration	subConfigurationNamed:	SimpleConfiguration
SCConfiguration	subConfigurationNamed:	SimpleConfiguration

Filter...

Flat Hier. | Current image SimpleConfiguration SCConfiguration | Class refs. Implementors Senders

subConfigurationNamed: aString

```
"I return a configuration of the name 'aString'. If this configuration doesn't exist i raise an error."

^ self configurations
  at: aString
  ifAbsent: [
    self configurations do: [ :conf | (conf subConfigurationNamed: aString) ifNotNil: [ :subsub | ^
subsub ] ].
    self error: 'No configuration named "', aString, '"' ]
```

7/10 [44] -- 7/10 [66] Format as you read W +L
Updating critiques...

Iceberg

- Paid by the consortium
- libgit2



Iceberg repositories

Repositories ☀ New repository + Clone repository + Add local repository ↶ Fetch all

Name	Current branch	Loaded version	Status
iceberg	dev-0.4	fefadb2 (9:40:15 am)	Up to date
libgit2-pharo-bindings	development	2a86f76 (19 April 2017)	Up to date
themes	master	1cb8ba9 (8 May 2017)	Up to date
* pharo	master	e9d735b (11 May 2017)	Uncommitted changes
TestToDelete	master	03bd70e (11 May 2017)	1 not published

General Packages Remotes Branches

Name	Value
Origin	git@github.com:pharo-vcs/iceberg.git
Remote User	
Remote host	github.com
Repo Owner	pharo-vcs
Location	/Users/esteban/Dev/Pharo/xx/pharo-local/iceberg/pharo-vcs/iceberg
Pulling from	origin
Pushing to	origin

Iceberg

- Potentially for

- Fossil



- Mercurial





Bitbucket



github
SOCIAL CODING



GitLab

Pharo on github



github
SOCIAL CODING

About classes

Class definition explosion

- subclass:, variableSubclass:, variableByte.. variableWord, weak:
- uses:
- layout:
- tag:
- immediate:, ephemeron
- category:
- package:
- slot:
- instanceVariableNames:, classVariableNames:

Need a fluid API

- Only specify what is used!
- No empty classVariableNames: “
- Handling combinatorial explosion

```
Class <<< #Box  
  vars: { #width . #height };  
  package: #MyPackage.
```


Possible Designs

```
Class {  
  superclass: #MySuperclass;  
  uses: #MyTrait;  
  vars: { #a. #b };  
  classVars: { #A. #B };  
  package: #MyPackage;  
  tags: #(Core).  
}
```

```
MySuperclass <<< #MyClass  
  uses: #MyTrait;  
  vars: { #a. #b };  
  classVars: { #A. #B };  
  tags: #(Core) ;  
  package: #MyPackage
```

Need a class ****definition****

- a.k.a. AST
 - traits, slots, kinds.....
- Can be manipulated
- Support for Smart Suggestions
- Need to be able to have ****undefined**** classes
- Important for modules!

Working on a “ClassParser” + Nodes

- Can handle variations, backward compatibles....
- Support for smart suggestion in class definition
- Avoid ‘*subclass:*’ matches: str

Opal

- Warnings into real objects
- Another pass on the API

PharoPro

More services around the consortium

- Long Term Support (LTS)
- Platform Support: CentOS, NetBSD, FreeBSD
- Pharo Hotline
- Custom Development

Pharo 70 is exciting

We are excited by the possibilities

You can help, contribute and have fun with us