



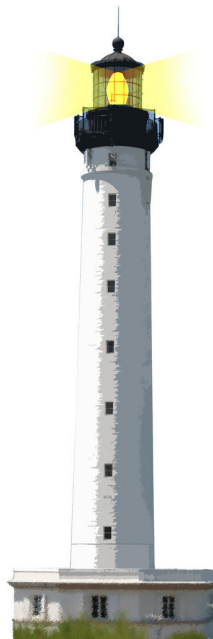
Learning Object-Oriented Programming and Design with TDD

Inheritance and Lookup

4: doesNotUnderstand: aMessage

Stéphane Ducasse

<http://stephane.ducasse.free.fr>



Goal

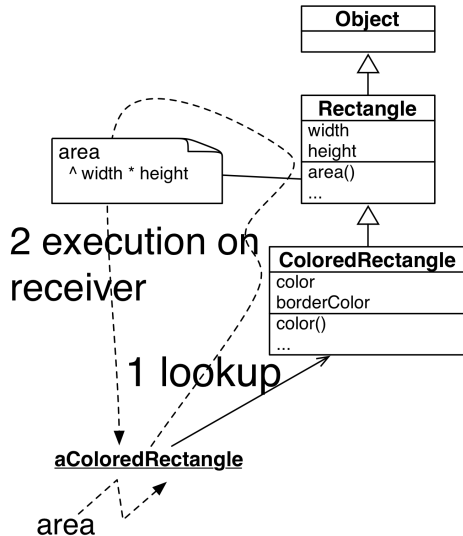
- A hook message is sent when the lookup fails
- Classes can customize the reaction to this failure



Message Sending

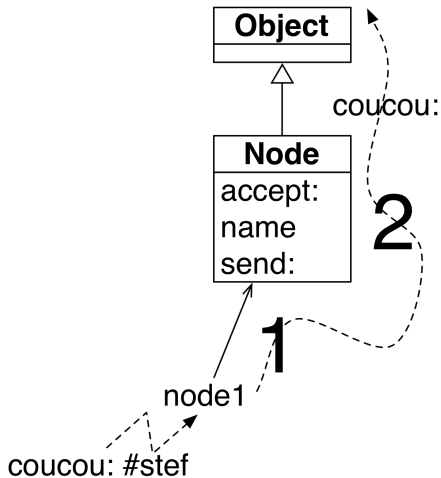
Sending a message is a two-step process:

1. **look up** the **method** matching the message
2. execute this method on the **receiver**

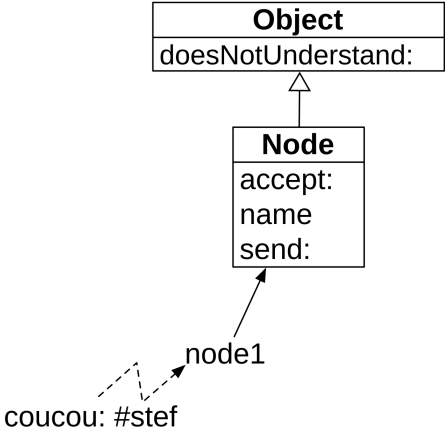


When No Method is Found

- if no method is found in the hierarchy, `doesNotUnderstand:` is sent to the receiver
- the initial message is passed as parameter



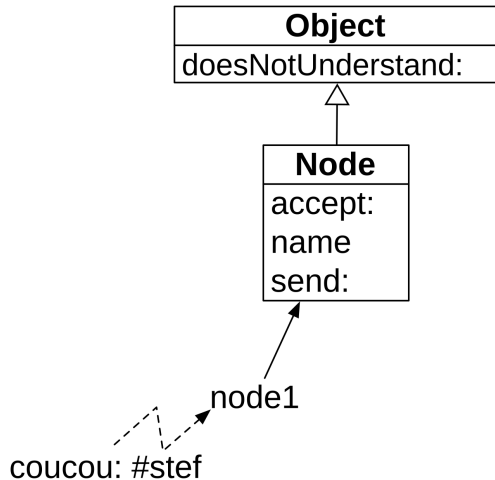
When No Method is Found



Step by Step...

node1 coucou: #stef

1. coucou: **looked up** in Node
2. **not defined** in Node -> **continues** in Object
3. **not defined** in Object -> **send**
doesNotUnderstand: **to** node1
4. doesNotUnderstand: **looked up** in Node
5. **not defined** in Node -> **continues** in Object
6. doesNotUnderstand: **is found** and executed



doesNotUnderstand: is a Message

- doesNotUnderstand: is a message
- any class can implement the corresponding method
- used for automatic delegation, proxy, distribution



doesNotUnderstand: and the Debugger

When no class redefines `doesNotUnderstand:`:

- `doesNotUnderstand:` in `Object` is **executed**
- this method **raises** a `MessageNotUnderstood` **exception**
- when not handled (by a `try catch` exception), an unhandled exception is raised
- this unhandled exception opens a debugger

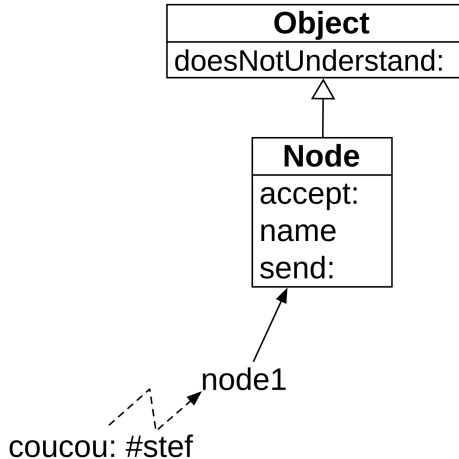


Catching MessageNotUnderstood Error

MessageNotUnderstood is just an exception, code can handle it:

```
Node >> sayHello
self coucou: #stef

Node >> welcome
[ self sayHello ]
on: MessageNotUnderstood
do: [ Transcript show: 'Something went wrong
with a message' ]
```



What You Should Know

- The lookup sends `doesNotUnderstand:` when it can not find a method
- Each class can implement this method
- The default implementation of `doesNotUnderstand:` raises an exception that can be handled



Resources

- Pharo mooc - Videos W4S04: <http://mooc.pharo.org>
- Pharo by Example: <http://books.pharo.org>



A course by Stéphane Ducasse
<http://stephane.ducasse.free.fr>

Reusing some parts of the Pharo Mocc by

Damien Cassou, Stéphane Ducasse, Luc Fabresse
<http://mocc.pharo.org>



Except where otherwise noted, this work is licensed under CC BY-NC-ND 3.0 France
<https://creativecommons.org/licenses/by-nc-nd/3.0/fr/>