Delegation of actions and accumulator

Form validation as an example

S. Ducasse, L. Fabresse, G. Polito, and P. Tesone

http://www.pharo.org
Objectives

- Think about objects
- Think about structure traversal
- Look at objects as accumulators
The case: Validation

- We want to validate UI forms
- Nested components may want to validate **or not** their contents
  - at input field or just at the pane level
Questions

- How can we navigate a tree of instances (widgets)?
- Where children can decide to be skipped?
- What do we report?
Validation

What should return the validation?

- Yes/no?
- Specific objects with semantics
  - e.g. filepath is isValid?
A formular: A tree of instances
A first design

- Any presenter can validate its contents
- Per default does nothing

```
SpPresenter >> isValid
^ true

SpPresenter >> report
^ OkReport new
```
A given item can refine it

MyFilePathPresenter >> isValid
^ self inputField isEndingBy: 'git'

MyFilePathPresenter >> report
^ WrongFileEndingReport new expecting: 'git' ; for: self path
A first design: Container

A container defines the semantics of collection for its children

SpOptionPresenter >> isValid
  ^ self children allSatisfy: [:each | each isValid ]

SpOptionPresenter >> report
  | report |
  self children do: [:childPresenter |
    childPresenter isValid ifFalse: [ report add: childPresenter report ]].
  ^ report
Flow’s first design

- Owner name: e.g., JohnDoe
- Project name: e.g., MyProject
- Local directory: /Users/ducasse/Workspace/FirstCircle/ActiveResearch
- Protocol: SSH

- e.g., JohnDoe
- e.g., MyProject
To have a report we need to know if the validation failed or not

Should `isValid` return a report?

If `isValid` returns a report then we have to return an ok report for anybody
Flow’s first design

Owner name: e.g., John Doe
Project name: e.g., My Project
Local directory: /Users/ducasse/Workspace/FirstCircle/ActiveResearch
Protocol: SSH

- report?
- ok?
- e.g., John Doe
- report?
- ok?
- e.g., My Project
- report?
- ok?
Second design: provide an accumulator

Pass around a basket and let any sub instance decide if it wants to participate

<table>
<thead>
<tr>
<th>Owner name</th>
<th>e.g., JohnDoe</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project name</td>
<td>e.g., MyProject</td>
</tr>
<tr>
<td>Local directory</td>
<td>/Users/ducasse/Workspace/FirstCircle/ActiveResea</td>
</tr>
<tr>
<td>Protocol</td>
<td>SSH</td>
</tr>
</tbody>
</table>
Second design: default

By default do not add to the report

SpPresenter >> validateInto: aReport
  ^ self
Second design: Containers and leaves

Each validating subcomponent

- gets the responsibility to fill up the report
- can bring its information to the report

MyFilePathPresenter >> validateInto: aReport
  ^ aReport add: (WrongFileEndingReport new expecting: 'git'; for: self path)

SpOptionPresenter >> validateInto: aReport
  ^ aReport
Conclusion

- Question interrogative forms
- Let the object decides if it wants to join a process but passing a container
- You may also have some double dispatch between the report and the container
- Explore design
Advanced Object-Oriented Design and Development with Pharo

A course by
S. Ducasse, L. Fabresse, G. Polito, and P. Tesone

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/