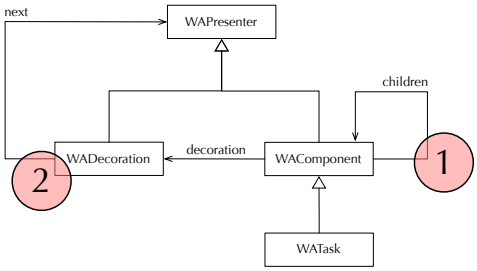
Presenters

Lukas Renggli

renggli@iam.unibe.ch, University of Bern
renggli@netstyle.ch, netstyle.ch GmbH

Overview



Subcomponents

- It is common for a component to display instances of other components.
- Components can be nested into each other using the composite pattern.
- A subcomponent is displayed using the method `#render:` in `WAHtmlRenderer`.

Initialize Children

- Subcomponents are usually stored within instance variables of the parent component.
- Subcomponents are commonly created lazily or as part of the components `#initialize` method.

```
SomeComponent>>initialize
    super initialize.
    myCounter := WACounter new.
```


Enable Children

- Parent components **must** implement a `#children` method that returns a collection of all of the subcomponents that it might display!
- If you fail to specify `#children` correctly, Seaside will raise an exception: *"Components not found while processing callbacks"*.

```
SomeComponent>>children
    ^ Array with: myCounter.
```


Render Children

- Children are rendered by sending the message `#render:` to the renderer.
- Never directly send `#renderContentOn:` to the subcomponent.

```
SomeComponent>>renderContentOn: html
    html heading: 'My Counter' level: 3.
    html render: myCounter.
```

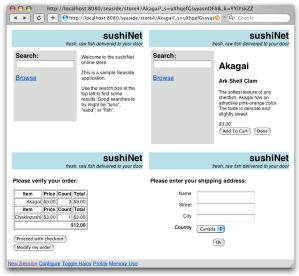
Replace Children

- Replace the subcomponent with a different instance of a component.
- Always register component for backtracking holding children that get replaced.

```
SomeComponent>>reset
    myCounter := WACounter new.
```


Multiple Control Flow

http://www.iam.unibe.ch/~scg/Archive/Papers/Duca04eSeaside.pdf



Decorations

- Decorations can be added to any component by calling `#addDecoration:`.
- Decorations are used to change the *behaviour* or the *look* of the decorated component.

Visual Decorations

- `WAMessageDecoration`
  Renders a heading above the component.
- `WAFormDecoration`
  Renders a form with buttons below the component.
- `WAWindowDecoration`
  Renders a border with a close button around the component.

Behavioral Decorations

- `WAValidationDecoration`
  Allows the validation of the answer-argument and display an eventually raised errors.
- `WABasicAuthentication`
  Protects a component with basic authentication.
- `WASessionProtector`
  Prevents the session being hijacked from another computer.

Internal Decorations

- `WATransaction`
  Used to implement `#isolate:`
- `WADelegation`
  Used to implement `#call:`
- `WAAnswerHandler`
  Used to handle `#answer:`

Interception

- #renderContentOn: to emit XHTML around the decorated component. Call #renderOwnerOn: to let the owners emit their output.
- #processChildCallbacks: to intercept the callback processing of the owners.
- #handleAnswer: to intercept the answer processing.

Examples

- AlignDecoration – Align a component visually to the left, the center or the right.
- PerformanceDecoration – Measure the rendering- and callback-preformance of a component.

```
SomeComponent>>initialize
    super initialize.
    self addDecoration: AlignDecoration new.
    self addDecoration: PerformanceDecoration new.
```


Align Decoration

```
AlignDecoration>>initialize
    super initialize.
    self beCentered.

AlignDecoration>>beOnTheLeft
    self align: 'left'.

AlignDecoration>>beCentered
    self align: 'center'.

AlignDecoration>>beOnTheRight
    self align: 'right'.

AlignDecoration>>renderContentOn: html
    html attributeAt: 'align' put: self align.
    html div: [ self renderOwnerOn: html ].
```

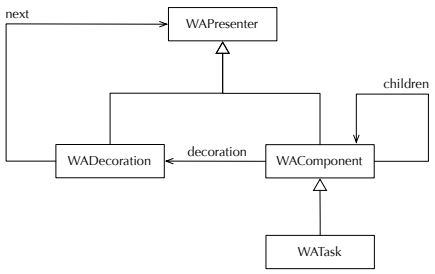

Performance Decoration

```
PerformanceDecoration>>renderContentOn: html
    | timing |
    timing := [ self renderOwnerOn: html ] timeToRun.
    self
        logRequest: html request
        response: WAResponse new
        kind: #render
        time: timing.
```


Performance Decoration

```
PerformanceDecoration>>processCallbackStream: aStream
    | timing continuation |
    timing := Time millisecondClockValue.
    continuation := self session escapeContinuation.
    self session escapeContinuation: [ :response |
        self
            logRequest: self session currentRequest
            response: response
            kind: #callback
            time: Time millisecondClockValue - timing.
        continuation value: response ].
    ^ super processCallbackStream: aStream.
```


Summary



Further Reading

- David Shaffer, A Seaside Tutorial
  - Embedding Components: http://www.shaffer-consulting.com/david/Seaside/
- Lukas Renggli, Seaside Tutorial:
  - Exercise: 30 – 31, 34