



Catching the Web Flow



Have you got the Flow?

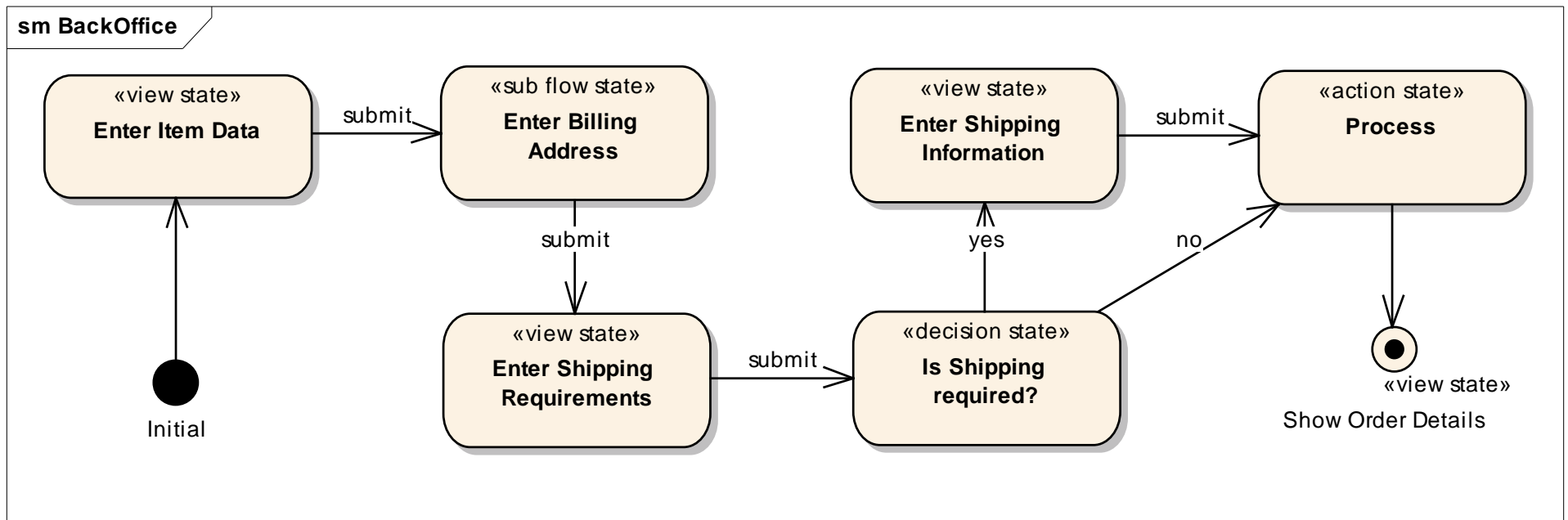
Björn Beskow

The Problem

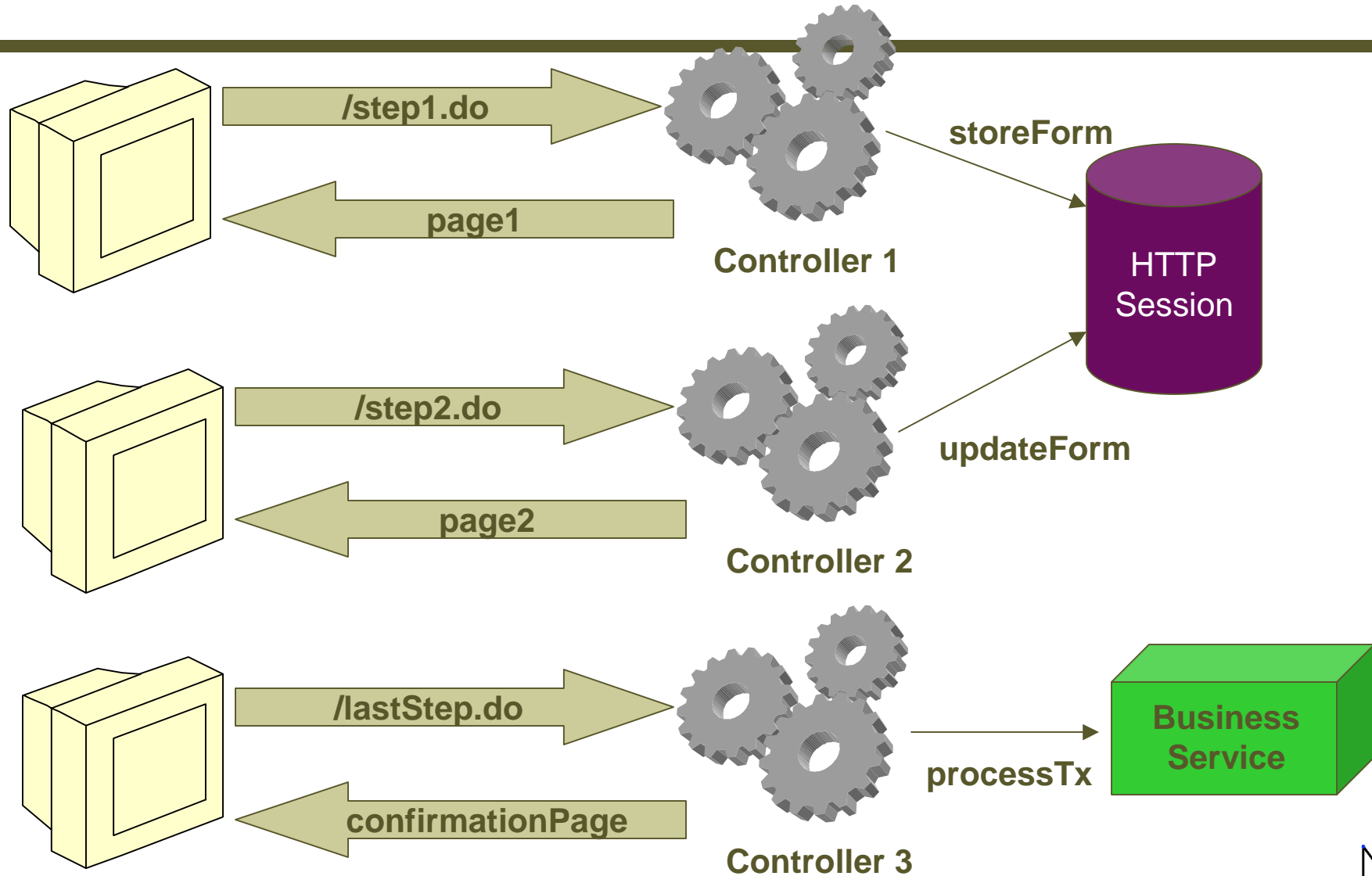
- Web Development is still messy
 - Request-centric, low-level
 - Difficult to express overall structure
 - “Application transactions” that spans several steps are particularly difficult
 - Browser issues a pain
 - Reuse is a pain



The "BackOffice" Flow Definition

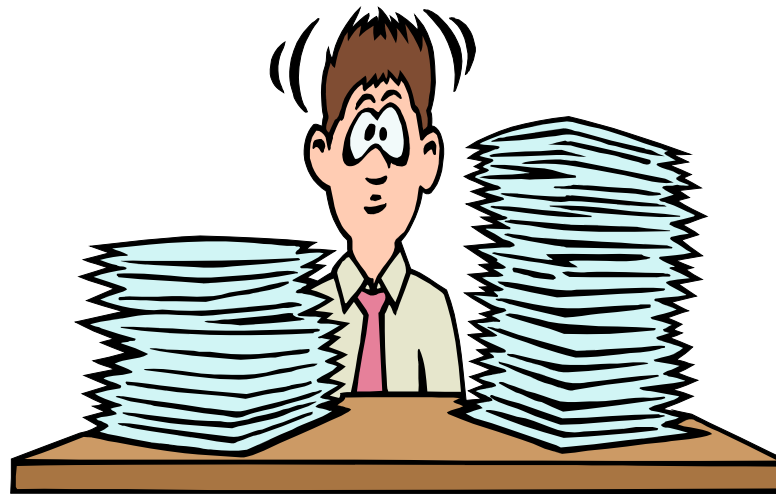


Current, Struts-based approach



Consequences

- Many lines of custom code are written to capture the interaction
- As an application grows more complex maintainability breaks down
- Fundamentally, a key abstraction is missing: the Flow
- A Flow is typically longer than a single request but shorter than a session, i.e. a conversation

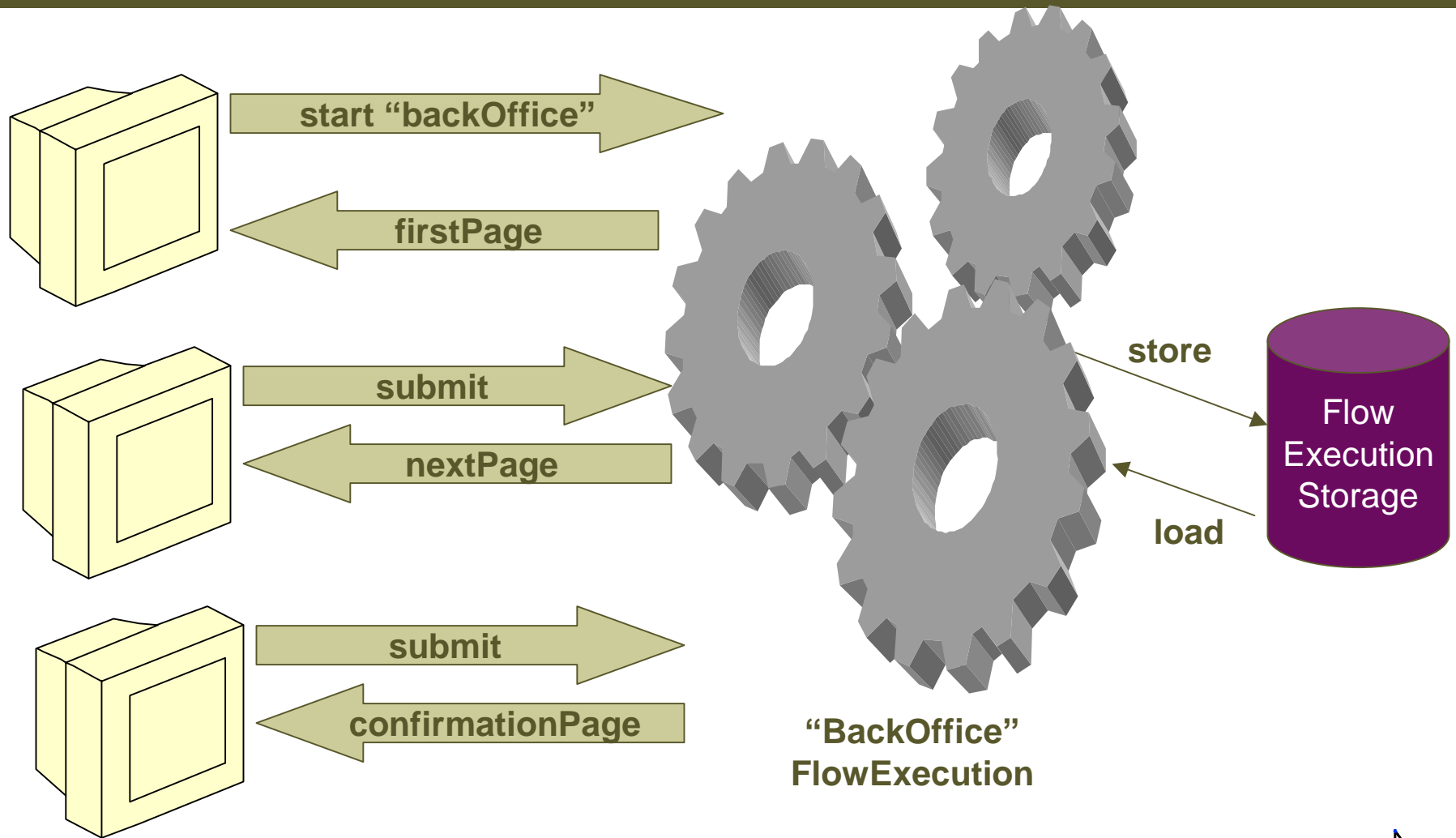


Spring Web Flow

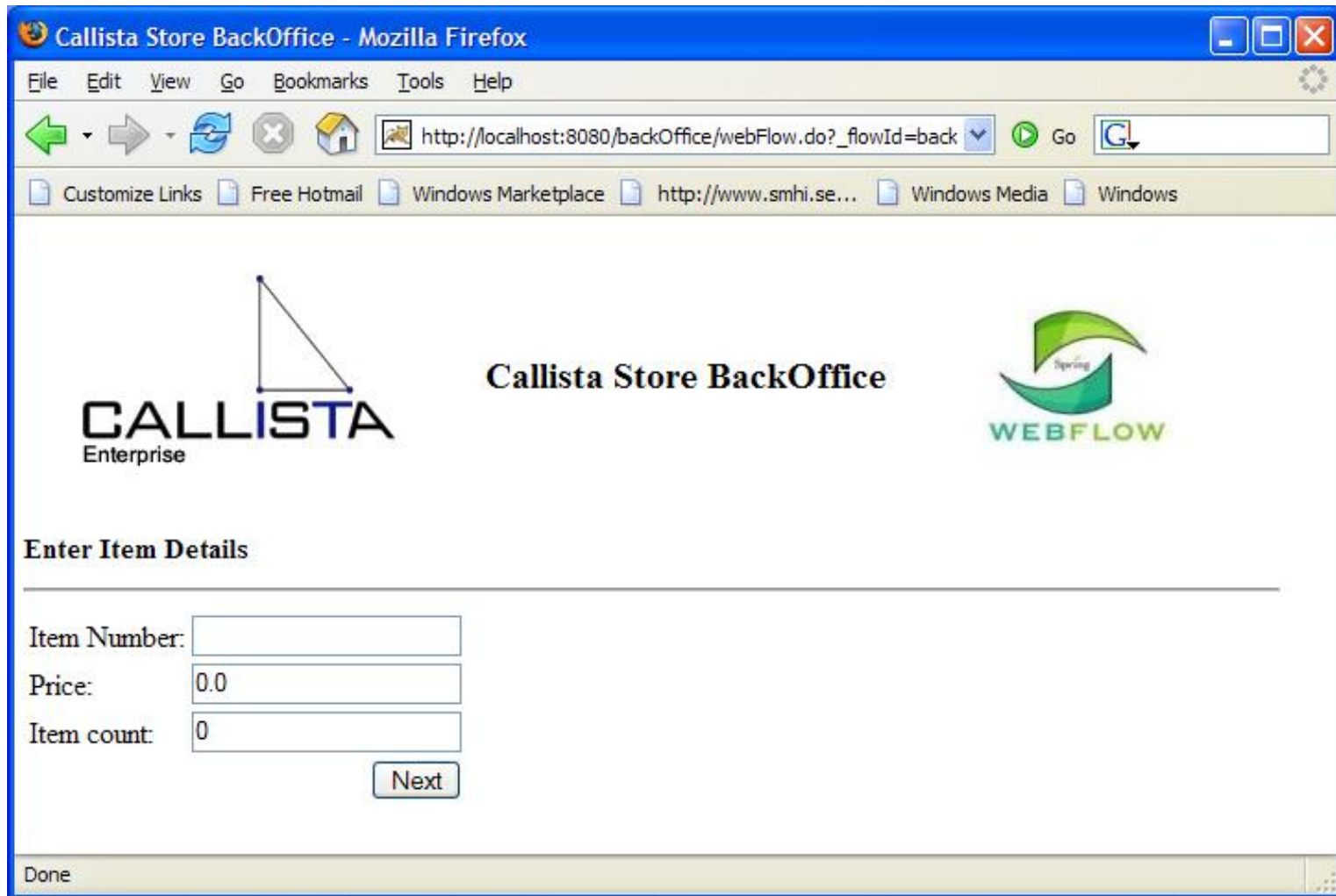
- Spring sub-project founded in 2005, based on Ervacon Web Flows
- Adds the missing abstraction: *The Flow*



Spring WebFlow approach

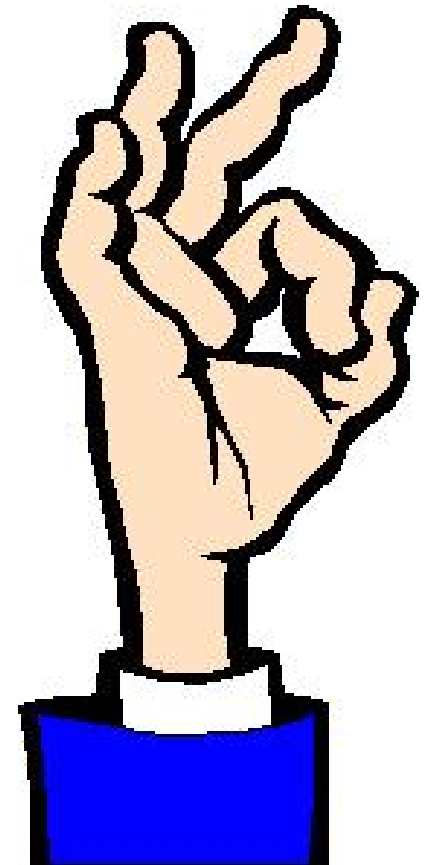


Demo



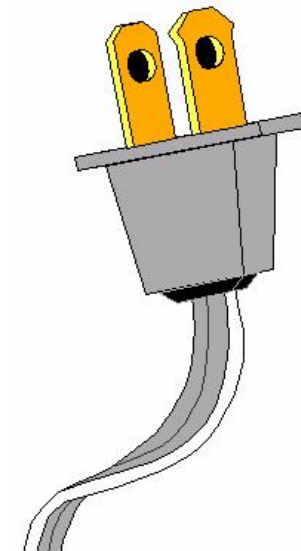
Advantages

- The logical flow is clearly defined, readable
- Navigation rules are encapsulated in one place
- Navigation is strictly enforced
- State management is automatic
- Controller logic is clearly decoupled from command logic
- HTTP independent
- No dependency on any request URLs



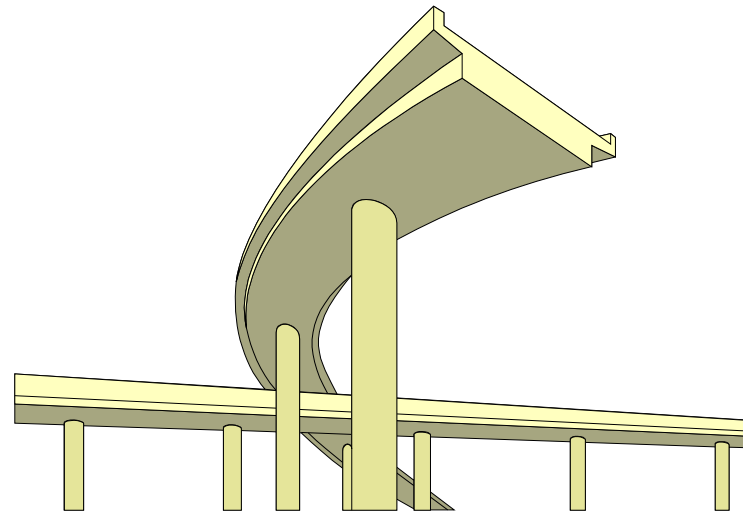
Integration with Web frameworks

- Integrating into other environments
 - Create a specialized FlowController for that environment
 - The FlowAction for Struts
 - The FlowNavigationHandler for JSF
 - The PortletFlowController for Portlets
- Spring Web Flow is not tightly coupled with any web framework



Spring WebFlow Road Map

- First official release for production use scheduled for Q1 2006
- Version 1.x enhancements
 - Tighter JSF integration
 - JMX-enabled flow management
 - Ajax support (Asynchronous View State Actions)
- www.springframework.org



Time for Questions!

