Using Struts 2 and Spring Frameworks Together

Bruce Phillips

University of Kansas

March 2009

Presentation and code examples available at
http://www.brucephillips.name/spring

References

- Spring Framework - http://www.springsource.org/documentation
- Spring Community Forums - http://forum.springsource.org/
- Introduction to the Spring Framework 2.5 - http://www.theserverside.com/tt/articles/article.tss?l=IntrotoSpring25
- Spring in Action, 2nd Edition, Manning Publishing, August 2007
- Pro Spring 2.5, Apress Publishing, August 2008
- Struts 2 Framework - http://struts.apache.org

References

- Using Struts 2 and Spring Frameworks Together, Bruce Phillips Blog - http://www.brucephillips.name/blog/index.cfm/2008/10/17/Using-Struts-2-and-Spring-Frameworks-Together
- Introduction to the Java Spring Framework, http://www.brucephillips.name/blog/index.cfm/2009/1/29/Introduction-to-the-Java-Spring-Framework
- Struts 2 In Practice, Manning Publishing, http://www.manning.com/wannemacher/
- Apache Struts 2 Documentation, Spring Plugin - http://struts.apache.org/2.x/docs/spring-plugin.html
- Struts User Forum - http://www.nabble.com/Struts---User-f206.html
- Maven - http://maven.apache.org/
- Maven - http://books.sonatype.com/maven-book/index.html

Spring and Struts 2 Frameworks

- Introduction to the Java Spring Framework
  - http://www.brucephillips.name/blog/index.cfm/2009/1/29/Introduction-to-the-Java-Spring-Framework
- Introduction to The Struts 2 Java Web Application Framework
  - http://www.brucephillips.name/blog/index.cfm/2008/10/7/Introduction-toThe-Struts-2-Java-Web-Application-Framework

Advantages of Using Spring and Struts 2 Together

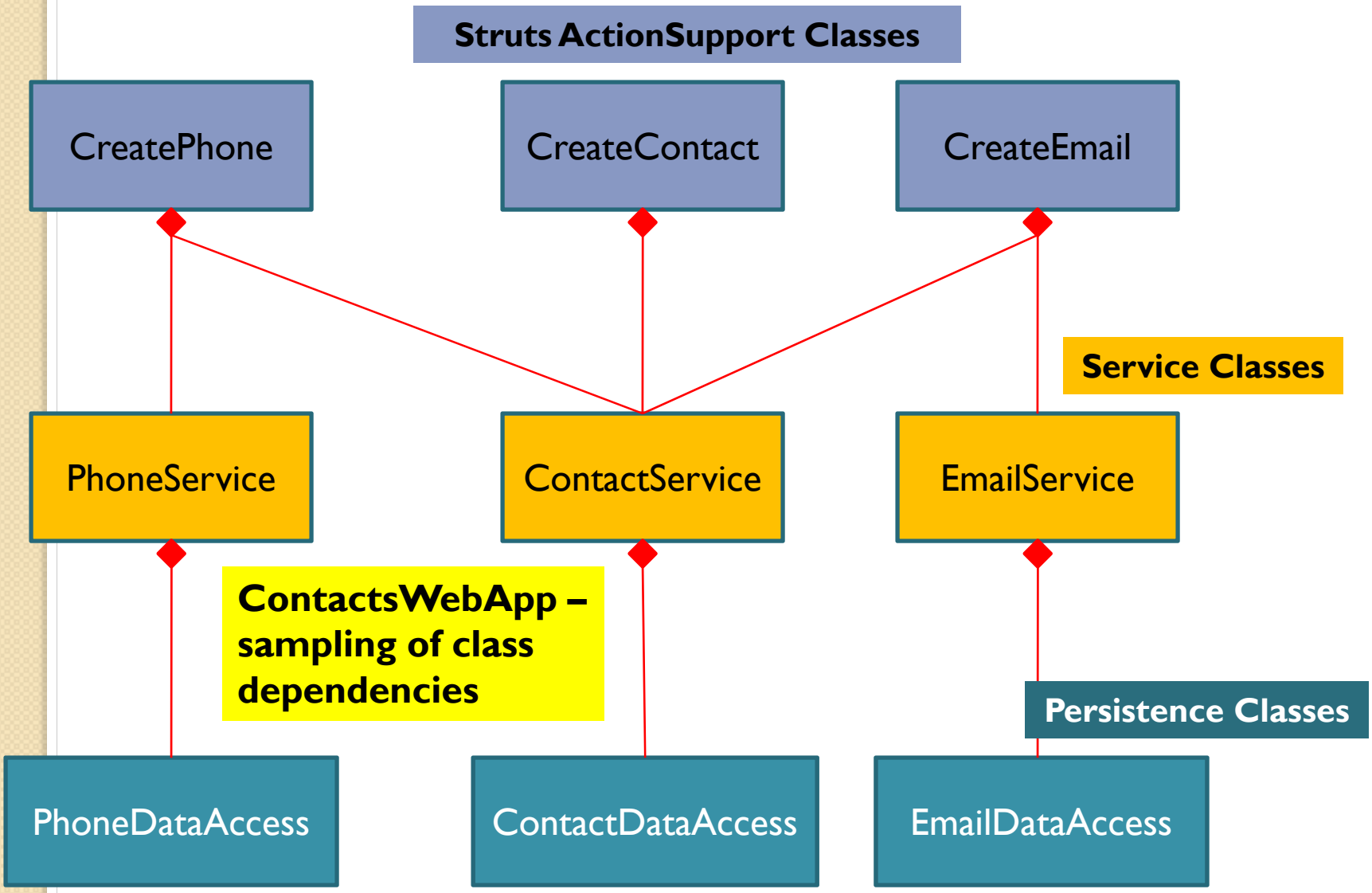
- Leverage the strengths of both frameworks
- Spring
  - Dependency management
  - Aspect-Oriented Programming (AOP)
  - Spring JDBC
- Struts 2
  - Simplifies implementing the model, view, controller (MVC) pattern in a web application

Example Applications

- Struts2_Spring_Example
  - Simple Struts 2 application that uses Spring to manage dependencies
- ContactsWebApp
  - More complicated Struts 2 application that uses Spring to manage dependencies, implement cross-cutting (AOP) concerns, and interact with a database repository
- Code and installation instructions at
  - http://www.brucephillips.name/spring

Classes Depend On Other Classes

Developers must "manage" these class dependencies (coupling)



Struts Without Spring

- Struts web applications tend to have quite a few "controller" (ActionSupport) classes
- These "controller" classes call upon other classes to do the work
- Without Spring must use concrete object instantiation
  - ContactService contactService = new ContactServiceImpl();
  - The same code will be repeated in all the other Struts ActionSupport classes that use a ContactService object
- Each class is responsible for managing its dependencies
  - More difficult to configure, maintain, and change

Struts With Spring

- Use Spring to manage dependencies
  - Dependencies are specified and managed in one place
  - Much easier to change dependencies
  - Most dependencies can be "auto-wired"
- Use the other benefits Spring provides
  - Aspect Oriented Programming (AOP)
  - Spring JDBC
  - Transaction Management

Jars Required

- See Struts2_Spring_Example application
  - jars are under Referenced Libraries
  - spring.jar is needed to use Spring capabilities
  - struts2-spring-plugin-2.1.6.jar is needed so that Struts 2 can use Spring

How To Integrate Spring Into Struts

- Changes to web.xml
  - Add ContextLoaderListener and RequestContextListener to web.xml
- Add applicationContext.xml
  - Under WEB-INF
  - Spring configuration and bean definitions

Use Spring to Manage Dependencies

- There are two different ways to have Spring manage the dependencies in a Struts application
  - Configure your actions as Spring beans
    - applicationContext.xml
  - Configure your actions in struts.xml
    - applicationContext_alternate.xml
  - In both cases use Spring to manage dependencies between your ActionSupport classes and other classes

Struts2_Spring_Example

- Register class is a Struts ActionSupport class
  - Register class depends upon a PersonService class
- Option 1 use Spring to create and manage the Register objects
  - See applicationContext.xml
    - Note the id value for the Register bean
  - Register class requires a PersonService object and Spring will manage this dependency
  - In struts.xml create the action node but for the class value use the id value specified in applicationContext.xml

Struts2_Spring_Example

- Option 2
  - See struts_alternate.xml
    - Create action node as usual by specifying the complete class for the class attribute
  - See applicatonContext_alternate.xml
    - Just create those beans for classes that are NOT Struts ActionSupport classes
  - Spring will manage any dependencies for the ActionSupport classes
    - Register class depends on PersonService class
    - Before a Register object is provided to the application, Spring will pass a PersonService object to class Register's setPersonService method

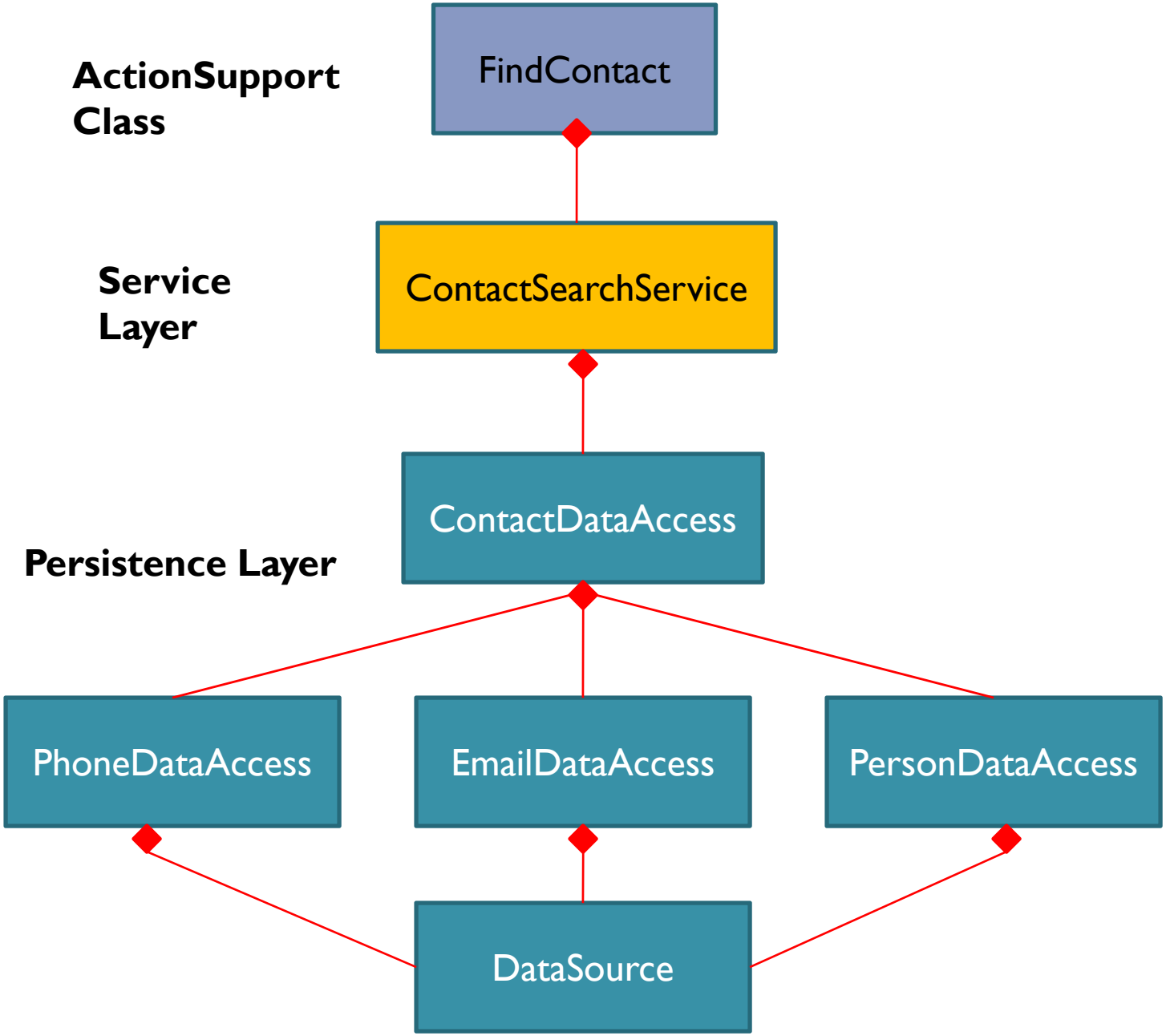
Struts 2 and Spring Configuration Options

- How both option 1 and option 2 manage dependencies can be configured to bypass the defaults
  - See chapter 3 Struts 2 In Practice
  - See Struts 2 Spring plugin documentation at:
    - http://struts.apache.org/2.x/docs/spring-plugin.html

ContactsWebApp

- Very simple contacts manager
- Web application that uses Struts 2 and Spring
  - Database is embedded Derby
- Uses Struts to manage ActionSupport classes (option 2) and Spring to:
  - Manage dependencies
  - Implement cross-cutting concerns such as logging, performance timing
  - Interact with database repository (Spring JDBC)

ContactsWebApp – FindContact and its dependencies



ContactsWebApp – struts.xml and applicationContext.xml

- struts.xml
  - Action nodes specify complete class value
  - For example FindContact
    - Note class FindContact needs a ContactSearchService object and has a public setContactSearchService method
- applicationContext.xml
  - Spring configuration
  - Create Spring beans for all non-ActionSupport classes that are used by the ActionSupport class

Struts 2 – Spring Interaction applicationContext.xml

- Spring will provide any objects needed by the ActionSupport classes automatically
  - Autowire "by name"
  - For example FindContacts property contactSearchService will get a value because Spring will call the setContactSearchService method and pass it a contactSearchService object
- Spring is also managing dependencies for the other classes
  - For example see contactSearchService, contactDataAccess and personDataAccess beans
  - Be sure to specify scope="request"

Spring AOP

- Can use Spring AOP just as we did in the non-Struts 2 Contact application
- Same configuration in the Struts 2 project's applicationContext.xml as we had in the Contact (non-web) project's applicationContext_DB_Repository.xml
- See package edu.ku.si.contact.advice

AOP Example - Exception Logging

- Need to specify where to apply the advice
  - See ExceptionLogging class PointCut definition
  - Don't need to apply advice to the ActionSupport classes because they won't be the originator of any Exceptions
- Rename c:/derby/contactsDB to c:/derby/tcontactsDB
- Run ContactsWebApp

Spring JDBC

- Can use Spring JDBC as we did in the Contact (non-Struts 2) application
- See package edu.ku.si.contact.data
- The data access classes extend Spring JDBC's SimpleJdbcDaoSupport
  - See PersonDataAccess
  - Much simpler to query database and process results
  - Note Spring JDBC will manage opening and closing connections, handling exceptions, etc.

Data Source Management

- The data access classes need a Data Source object
  - See applicationContext.xml
  - Note no scope is specified for the dataSource bean
- By extending SimpleJdbcDaoSupport the data access classes inherit the setDataSource method which Spring uses to inject the data source object
- To use connection pooling see notes in applicationContext.xml

Why Use Spring With Struts 2?

- Significantly reduces coupling between classes
- Central management of dependencies
- Applications are easier to maintain and change
- Leverage Spring features that Struts 2 doesn't provide
  - Aspect-oriented programming
  - Spring JDBC