Designing With Dependency Injection

Alex Miller
MetaMatrix

Overview

- Definition
- Types
- Consequences
- Patterns / Smells
- Architecture
- Q&A

Definition

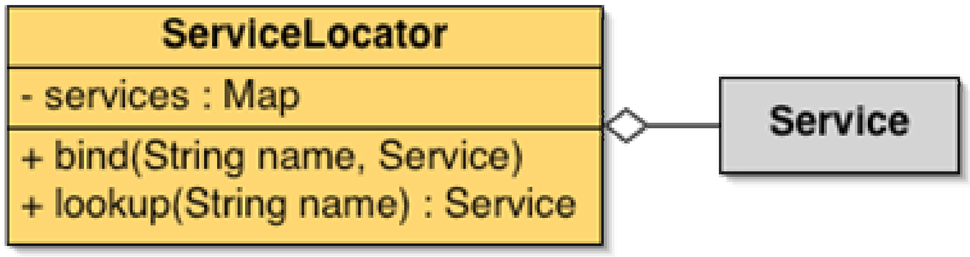
- Other terms
  - Inversion of Control
  - Hollywood principle
- My definition
  - Technique to reduce coupling by moving configuration and dependency wiring outside a component
- Configuration as dependency

Types

- Service locator
- Interface injection (Type 1)
- Setter injection (Type 2)
- Constructor injection (Type 3)
- Getter injection
- Context IOC?

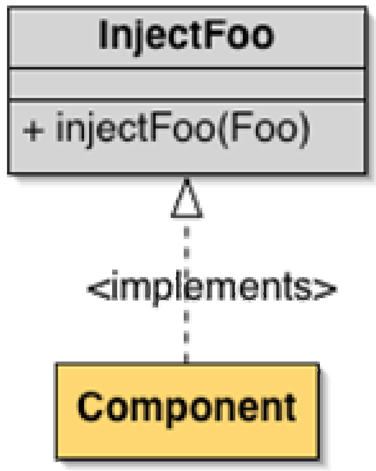
Service locator

- A facility to bind and look up components by name or type
- Not dependency injection (component still controls the wiring)
- Example: JNDI



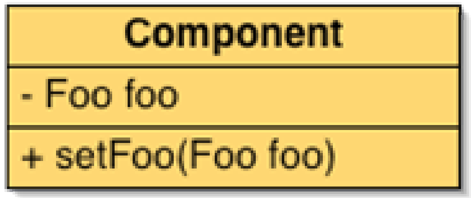
Interface Injection (Type 1)

- Define interface used to inject a dependency
- Container: Apache Avalon



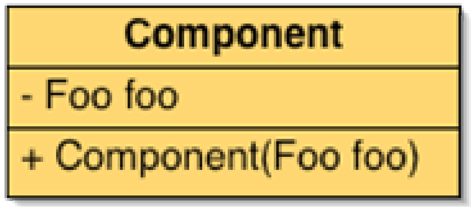
Setter Injection (Type 2)

- All dependencies declared using setter methods
- Probably the most common type
- Containers: Spring, Pico



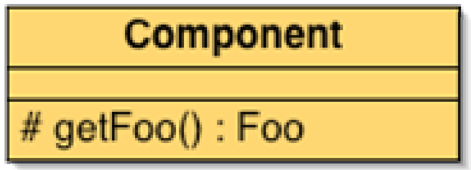
Constructor Injection (Type 3)

- Declare dependencies completely in one or more constructors (if different possible sets)
- Next most common after Type 2
- Containers: Pico, Spring



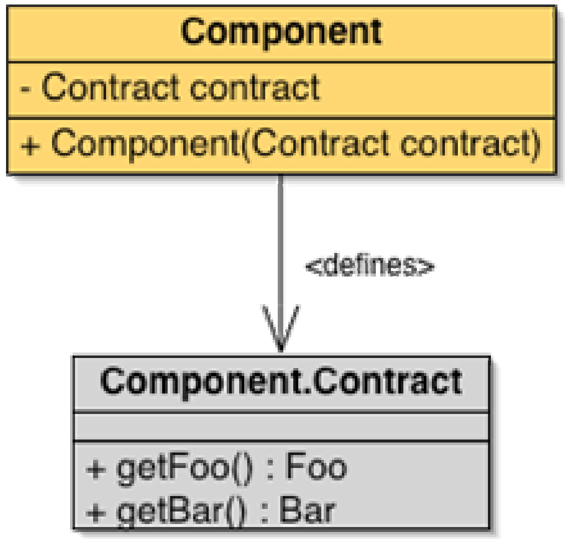
Getter Injection

- Specify a getter to be used within the class to retrieve the dependency
- Use AOP to inject a dependency by changing the implementation of the getter OR subclass and override the method
- Example: proposed by Bob Lee and prototyped using dynaop



Context IOC

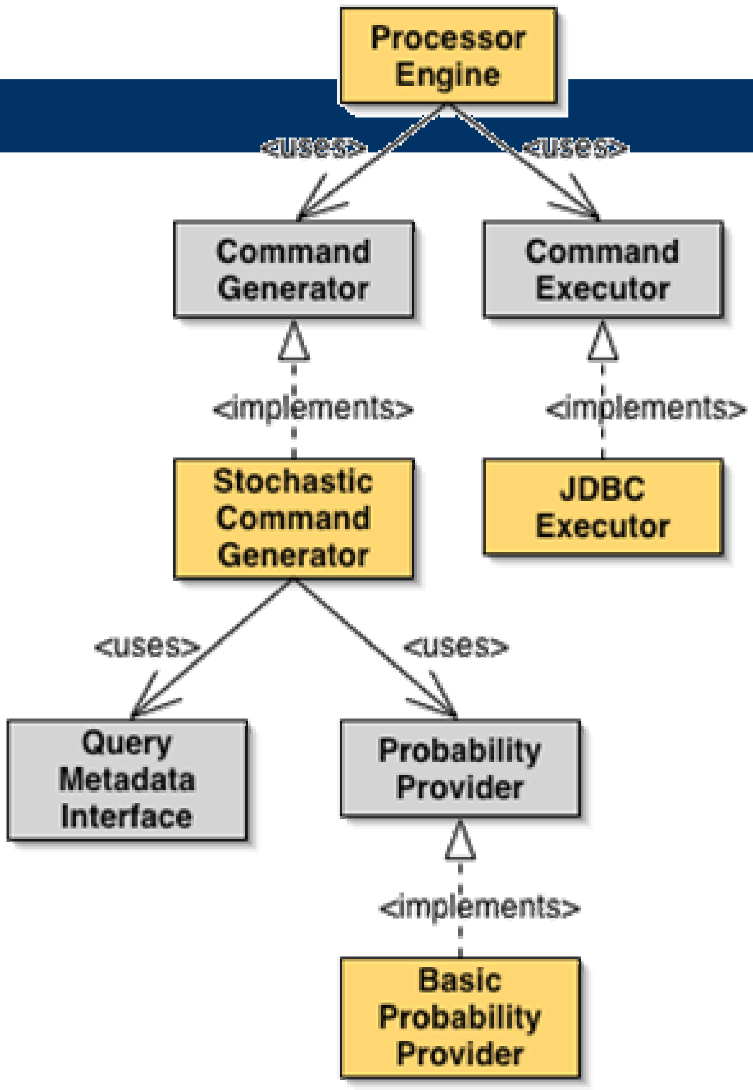
- Proposed this week on TheServerSide.com
- Component defines an inner interface with getters for all dependencies
- Constructor takes an instance of this interface



Consequences

- Flexible
- Testable
- Maintainable

Example



Processor Engine
<uses>
<uses>
Command Generator
Command Executor
<implements>
<implements>
Stochastic Command Generator
JDBC Executor
<uses>
<uses>
Query Metadata Interface
Probability Provider
<implements>
Basic Probability Provider

Patterns

Category	Pattern
Base	Separated Interface
Base	Plugin
Base	Service Stub
Base	Registry
Creational	Factory Method
Creational	Builder
Creational	Scripted configuration
Creational	Declarative wiring
Structural	Adapter
Structural	Decorator / Proxy

Base Patterns

- **Separated Interface** - put interface in separate package than implementation
- **Plugin** - link classes during configuration rather than compilation
- **Service Stub** - removes dependence on a service during testing
- **Registry** - basically a service locator

Creational Patterns

- **Factory Method** - method to create concrete instances of an abstract type
- **Builder** - object used to configure and create concrete instances
- **Scripted configuration** - move component assembly and wiring to an interpreted script
- **Declarative wiring** - assemble components using a lightweight container and config file

Structural Patterns

- **Adapter** - adapts one interface to another
- **Decorator** - add behavior to an interface by wrapping it
- **Proxy** - change behavior by inserting a proxy

Smells

- Components without interfaces
- Untestable component
- Singletons
- Property hell

Architecture

- When to use service locators
- Open vs closed designs
- External APIs
- Pervasive dependencies (logging)

Links

- **Books**
- Fowler, Martin, Patterns of Enterprise Application Architecture, http://martinfowler.com/books.html#eaa
- Gamma, et al, Design Patterns, http://hillside.net/patterns/DPBook/DPBook.html
- **Blogs**
- Miller, Alex, http://www.jroller.com/page/metalex
- Fowler, Martin, Inversion of Control Containers and the Dependency Injection pattern, http://www.martinfowler.com/articles/injection.html
- Oberg, Rickard, Dependency injection and open vs. closed designs, http://jroller.com/page/rickard/20040814#dependency_injection_and_open_vs
- Lee, Bob, Getter-Based Dependency Injection, http://weblogs.java.net/blog/crazybob/archive/2004/05/getterbased_dep.html
- Beust, Cedric, Getter-Based Injection, http://www.beust.com/weblog/archives/000134.html
- Denny, Mitch, IOC and .NET http://notgartner.com/posts/906.aspx
- Cazullino, Daniel, Lightweight Containers and Plugin Architectures: Dependency Injection and Dynamic Service Locators in .NET, http://weblogs.asp.net/cazzu/archive/2004/05/10/129140.aspx
- Weirich, Jim, Dependency Injection in Ruby, http://onestepback.org/index.cgi/Tech/Ruby/DependencyInjectionInRuby.rdoc
- Thomas, Dave, Transparent Inversion of Control, http://blogs.pragprog.com/cgi-bin/pragdave.cgi
- Mathew, Sony, Examining the Validity of Inversion of Control, http://stage.theserverside.com/articles/article.tss?l=IOCandEJB
- **Software Projects**
- Spring, http://www.springframework.org/
- Pico, http://www.picocontainer.org/
- Hivemind, http://jakarta.apache.org/hivemind/
- Needle for Ruby, http://needle.rubyforge.org/