## Use Scenarios of Design Patterns

Design Pattern Name Abstract Factory	Topic ID(s) 99	Summary use abstract factory to implement dependency injection	Online Materials
Abstract Factory	68	embedded in Ruby on Rails as ActiveRecord	I
Active Record	80 120	PostgreSQL specific usage of active record modified version of active record pattern in CodeIgniter	I I
Adapter	91 52	Android Adapter IBM MobileFirst/Worklight adapters	I I, II
Builder	43 7	Ruby XML Builder Java StringBuilder	I, II I
CQRS	115 118	use CQRS in Domain Driven Design Axon Framework	I I
Command	100 1, 48, 95	command pattern in game programming command pattern in GUI design (WPF, GWT, WinForms)	I, II I, II, III
	9 93	composite in GUI design (SWT, GWT, WPF, JSF) composite data entity	I, II, III, IV
Composite	71 43	composite in JavaScript frameworks (Marionette.js, ExtJS) composite for file systems	I, II I
Content Negotiation	114 58	content negotiation in ASP.NET content negotiation in Spring MVC	I
CRTP	32	crtp in C++	I
DAO	89 56	DAO in Microsoft Access combine DAO with service layer	I I
	62, 58 37, 68, 27, 103, 85	DAO in Spring framework  DataMapper library	I I
Data Mapper	56 93, 118, 115	data mapper in domain models DataMapper ORM library (involving Entity Framework, Hibernate, Doctrine)	I
Data Mapper	120, 64	PHP DataMapper	Ī
DTO	118 93	iBATIS DataMapper framework combine DTO with Entity Framework	I
	115 41	use DTO in domain riven design decorator in Python	I I
	106 60	decorator for registering in Django decorator in Zend Framework	I
Decorator	73	decorator for login in Django	Ĭ
	57 7	decorator in AngularJS decorator in TypeScript	I I
	62 14	dependency injection in Spring Bean Google dependency injection framework (Guice, Dagger)	I I, II
	57	dependency injection in Angular	I, II
Dependency Injection	12 94	dependency injection in ASP.NET Core dependency injection in PHP framework (Symfony, Laravel)	I I, II
	101 28	use dependency injection in WPF to decouple views dependency injection in AngularJS	I, II I
	56	dependency injection in multi-layer application  Domain Model in Domain Driven Design	I, II
Domain Model	115 93	domain model in ORM (Entity Framework, Hibernate)	I, II
Double Checked Locking	27 16	Grails Domain Class use double checked locking to implement thread-safe singleton	I I
Event Sourcing	76 8	combine event sourcing with CQRS event sourcing in Event Store database	I
	44	event sourcing in Apache Kafka	I
Facade	94, 64 56	facade in Laravel framework facade for services	I, II
Tucucc	62 114	EJB session bean facade facade for authentication	I I
Factory Method	16 62	Class Factory Methods in Objective-C use factory method to create Spring Bean	I
To at a second	57	factory in AngularJS Python Factory	I T
Factory	41 94	factory in Symfony	I
Federated Identity	73 52	federated identity in Microsoft Azure federated identity in Amazon Cognito	I I
File Transfer	44 64	file transfer between message queues front controller in PHP (Zend Framework, Symfony)	I I, II
Front Controller	58	Java implementation of front controller	I
Future	32 78	future in Scala future in C++	I I
HMVC	64 118	implement HMVC in CodeIgniter Kohana framework	I I
	58 106	Spring MVC Interceptor (HandlerInterceptor, WebRequestInterceptor) Castle Windsor Interceptor	I
	93	Hibernate Interceptor	I
Interceptor	73 57	login interceptor Angular http interceptor	I I
	69 67	Struts Interceptor Flume Interceptor	I
	78	C++ iterator, vector/list iterator	Ĭ
	53 43	Java iterator, ListIterator read file by iterator	I I
Iterator	75 107	traverse multidimensional data struct traverse tree struct, e.g., QTreeWidgetItemIterator	I
	97	C++ map iterator	I
	36		1
	36 64	Java iterator PHP DirectoryIterator	I
			I, II I, II
Lazy Loading	93 46 28	PHP DirectoryIterator entity lazy loading in ORM (Entity Framework, Nhibernate) images lazy loading in jQuery feature modules lazy loading in Angular	
Lazy Loading	93 46 28 91 69	PHP DirectoryIterator entity lazy loading in ORM (Entity Framework, Nhibernate) images lazy loading in jQuery feature modules lazy loading in Angular images lazy loading in Android ListView DataTables lazy loading in PrimeFaces	I, II
Lazy Loading	93 46 28 91	PHP DirectoryIterator entity lazy loading in ORM (Entity Framework, Nhibernate) images lazy loading in jQuery feature modules lazy loading in Angular images lazy loading in Android ListView	I, II
Lazy Loading  Master/Slave	93 46 28 91 69 8 52 37	PHP DirectoryIterator  entity lazy loading in ORM (Entity Framework, Nhibernate) images lazy loading in jQuery feature modules lazy loading in Angular images lazy loading in Android ListView DataTables lazy loading in PrimeFaces  MongoDB master-slave replication (database) Jenkins master/slave architecture (project management) MySQL master-slave replication (database)	I, II
	93 46 28 91 69 8 52 37 44 15	entity lazy loading in ORM (Entity Framework, Nhibernate) images lazy loading in jQuery feature modules lazy loading in Angular images lazy loading in Android ListView DataTables lazy loading in PrimeFaces  MongoDB master-slave replication (database) Jenkins master/slave architecture (project management) MySQL master-slave replication (database) using in ActiveMQ for high availabliity (message queue) Bluetooth master-slave model (communication)	I, II
	93 46 28 91 69 8 52 37 44	PHP DirectoryIterator  entity lazy loading in ORM (Entity Framework, Nhibernate) images lazy loading in jQuery feature modules lazy loading in Angular images lazy loading in Android ListView DataTables lazy loading in PrimeFaces  MongoDB master-slave replication (database) Jenkins master/slave architecture (project management) MySQL master-slave replication (database) using in ActiveMQ for high availabliity (message queue)	I, II I I I I I I I

Mediator	114, 43, 52 48 1	mediators in WSO ESB event mediator mediator in MVVM	I I I
Message Broker	44 89	various message brokers (RabbitMQ, ActiveMQ, Kafka, WSO2 Message Broker) WebSphere Message Broker	I, II, III, IV I
	44	use message queue to implement messaging	I
Messaging	76 $22$	event-driven messaging messaging in multithreading	I I
	120, 64	PHP MVC framework (CakePHP, Zend Framework, CodeIgniter)	I, II, III
MVC	23 58	ASP.NET MVC (Routing) Spring MVC	I, II I
MVC	48	MVC in Java GUI design (JavaFX, Swing)	I, II
	71 85	JavaScript MVC architecture (Knockout.js, Node.js, AngularJS) Ruby on Rails MVC framework	I, II, III I
	39, 91	MVP in Android development	I
MVP	14 $1, 92, 102$	combine MVP with Dagger 2 for dependency injection (Android) MVP in WinForms/WPF	I I, II
	102	constructing architectures in WPF	Ĭ
MVVM	101 71	MVVM Light Toolkit MVVM in KnockoutJS	I
	39	MVVM in mobile development (Android, iOS)	I, II
	69 8, 27, 71	applying MVVM to Kendo UI ObjectId in MongoDB (Mongoose, Meteor)	I
Object ID	68	object_id in Ruby	I
	39 100	objectId in Parse avoiding memory fragmentation in game programming	I
Object Pool	91	avoiding memory fragmentation in game programming reusing ListViews in Android development	I
	39	observer in mobile development (Android, iOS)	I, II
Observer	103 90	observer in Ruby on Rails observer in Magento	I
	110	page object in Selenium	I
Page Objects	53	page object in Watir	I
J • ···	28 37	page object in Protractor use page object with RubyGems	I
	41, 43	shell script pipeline (Unix shell, Powershell, Bash)	I, II, III
	75 116	pipeline for machine learning (Scikit-Learn)	I
Din alin	116 52	pipeline in MIPS architecture pipeline for projects Continuous Integration/Delivery (Jenkins)	I
Pipeline	42	graphics pipeline (OpenGL, DirectX)	I, II
	44 8	pipeline for web service (NServiceBus, Redis, BizTalk) pipeline for data processing (MongoDB, Hadoop)	I, II, III I, II
	23	pipeline for JavaScript and CSS assets	I, 11
	119	http connection pooling	I
Pooling	22 100	thread pooling object pooling in game design	I
	112	database connection pooling	I
	<u>44</u> 58	object pooling in communication  Post-Redirect-Get in Spring MVC	I
Post/Redirect/Get	120	Post-Redirect-Get in ASP.NET MVC	I
	71	use Publish/Subscribe in JavaScript	I
Publish/Subscribe	15 8	use Publish/Subscribe in Android Redis Publish/Subscribe	I
	80	Meteor Publish/Subscribe	Ī
Reactor	62 48	Spring 5 Reactor Python Twisted Reactor	I
Record Set	89	Recordset in MS ADO (Access, Excel)	I, II, III
	7	use reflection to handle generic types	I
Reflection	36 6	Java Reflection C# Reflection	I
	12	DbContext: combine unit Of work and repository (Entity Framework)	I
	56, 104, 93	use repository in Entity Framework with ASP.NET	I
Repository	115 106	use repository in Domain-Driven Design (Aggregates) use repository with dependency injection	I
	94	repository in Laravel Doctrine	I
	20 56, 12	generic repository  combine service layer with data access layer (repository, entity)	I, II
Service Layers	62	Spring bean as service layer	I, II I, II
-	73	service layer for MVC application	I, II
Service Locator	99 106	use service locator with dependency injection use service locator in Unity Container	I I
	64	service locator in Zend Framework	Ï
	8 27, 119	database sharding (MongoDB) database sharding (MySQL, PostgreSQL)	I I, II
Sharding	27, 119 44	message queue sharding (RabbitMQ)	I, 11
	76	Akka cluster sharding	I
STI	103, 86, 85, 27 93	single table inheritance in Ruby on Rails single table inheritance in ORM (Hibernate ORM, Entity Framework, Doctrine)	I I, II, III
	119	singleton class for database connection (bad design)	I
Singleton	62 54	Spring Bean singleton scope	I
	54 41	singleton UIViewController in iOS development (bad design) singleton-decorator in Python	I, II I
Y	71	React component state	I
State	100 23	using state pattern in game programming state pattern for routing in Angular UI.Router	I
	22	throttling on threads	I, II
Γhrottling	44	throttling on web services	I, II
Ü	119 46	WCF throttling throttling on resize/scroll events	I I, II
	12	unit of work with repository pattern	Ι
Unit of Work	93, 56, 104	unit of work in ORM (Entity Framework, Nhibernate)	I, II
	106 115	unit of work with Unity Container unit of work in Domain Driven Design	I I
Value Object	115, 56, 99	value object in domain driven design	I, II
	93	value object in ORM (Nhibernate, Entity Framework)	I, II
Viewcontroller	54, 39, 42 107	UIViewController in iOS  combining with the traversal strategies of tree structure	I, II
Visitor	36	parsing Abstract Syntax Trees	I, II
	43, 60	transforming structures into xml files	I

## Related Design Pattern Pairs

Design Pattern Pair   Category   Summany   Online Mater
Abstract Factory - Factory Method Abstract Factory - Pactory Active Record - DAO Analogy Active Record - Dath Active Record - Repository Active Record - Single Table Inheritance Adapter - Pacade Builder - Facade Analogy Builder - Pacade Builder - Decorator Analogy Builder - Decorator Analogy Co-operation CQRS - Event Sourcing Co-operation CQRS - Event Sourcing Co-operation Command - MVVM Dependency DAO - Repository DAO - Pactory DAO - Pactory DAO - Repository DAO - Pactory DAO - Repository DAO - Re
Abstract Factory - Factory Method Abstract Factory - Factory Service Active Record - DAO Analogy Active Record - Data Mapper Active Record - Data Mapper Active Record - Laxy Loading Co-operation Active Record - Laxy Loading Active Record - Laxy Loading Co-operation Active Record - Laxy Loading Active Record - Laxy Loading Active Record - Single Table Inheritance Active Record - Singleton Active Record - S
Active Record - Data Mapper Active Record - Data Mapper Active Record - Data Mapper Active Record - Lazy Loading Active Record - WC Dependency Active Record - MC Dependency Active Record - Nepository Active Rec
Active Record - Data Mapper Active Record - Data Mapper Active Record - Lazy Londing Active Record - Repository Active Record - Repository Command - Strategy Command - Strategy Command - Strategy Command - Strategy Analogy Co-operation Composite - Visitor Composite - Visitor Data Mapper - Pomonin Model Data Mapper - Pomonin Model Data Mapper - Pomonin Model Data Mapper - Repository DTO - MVWM Dependency DTO - MVWM Dependency DTO - Service Layers Dependency Injection - Factory Dependency Injection - Reflection Dependency Injection - Reflection Dependency Injection - Reflection Dependency Injection - Strategy Double Checked Locking - Singleton Pactor - Singleto
Active Record - Data Mapper Active Record - Data Mapper Active Record - MVC Dependency Active Record - Sugale Table Indiritance Adapter - Pecorator Adapter - Decorator Adapter - Decorator Adapter - Strategy Analogy Builder - Decorator Analogy Analogy Analogy Coperation CQRS - Sevent Sourcing Cooperation CQRS - Sevent Sourcing Cooperation CQRS - Sevent Sourcing Command - MVVM Dependency Analogy Command - MVVM Dependency DAO - Record Set DAO - Pactory DAO - Record Set Data Mapper - Domain Model Co-operation DAO - Record Set DATE - Data Mapper - Domain Model Co-operation DAO - Record Set DATE - Data Mapper - Domain Model Co-operation DAO - Record Set DATE - Data Mapper - Domain Model Co-operation DAO - Record Set DATE - Data Mapper - Domain Model Co-operation DAO - Record Set DATE - Data Mapper - Domain Model Co-operation DAO - Record Set DATE - Data Mapper - Domain Model Co-operation DAO - Record Set DATE - Data Mapper - Domain Model Co-operation DAO - Record Set DATE - Data Mapper - Domain Model Co-operation DAO - Record Set DATE - Data Mapper - Domain Model Co-operation DAO - Record Set DATE - Data Mapper - Domain Model Co-operation DAO - Record Set DATE - Data Mapper - Domain Model Co-operation DAO - Record Set DATE - Data Mapper - Domain Model Co-operation DAO - Record Set DATE - Data Mapper - Domain Model Co-operation DATE - Data Mapper - Data Mapper - Domain Model Co-operation DATE - Data Mapper - Data Mapper - Domain Model Co-operation DATE - Data Mapper - Domain Model Co-operation DATE - Data Mapper
Active Record - Lazy Loading Active Record - NVC Active Record - Special Coperation Active Record - Single Table Inheritance Adapter - Decorator Adapter - Decorator Adapter - Strade Analogy Active Record - Single Table Inheritance Adapter - Strade Analogy Adapter - Strade Analogy Adapter - Strade Analogy Builder - Decorator Analogy Analogy Coperation CQRS - Sewn Sourcing Co-operation CQRS - Sewn Sourcing Co-operation CQRS - Sewn Sourcing Co-operation COMPS - Coperation DO - DTO Command - MVVM Dependency DAO - Ractory Coperation DAO - Pactory DAO - Ractory DAO - Ractory DAO - Record Set Data Mapper - Domain Model Co-operation DAO - Record Set DATE - Dependency Dipection - Repository DAO - Ractory Dependency Injection - Interceptor Dependency Injection - Reflection Dependency Injection - Service Locator Dependency Injection - Reflection Dependency Injection - Service Locator Analogy Dependency Injection - Reflection
Active Record - Rupostory Active Record - Rupostory Active Record - Single Table Inheritance Adapter - Decorator Analogy Builder - Rectory CQRS - Event Sourcing CQRS - Event Sourcing CO-operation CO-operation Command - MVVM Dependency DAO - Bactory DAO - Ratory DAO - Repository DTO - AWVM Dependency DTO - Service Layers DPO - Service Layers Dependency Injection - Factory Dependency Injection - Factory Dependency Injection - Factory Dependency Injection - Service Locator Peacules - Mediator Pactor - Service Locator Pactory - Service Locator Dependency Injection - Service Locator Pactory - Singleton Dependency Injection - Service Locator Pactory - Singleton Dependency Injection - Service Locator Pactory - Service Locator Pactory - Singleton Pactory - Service Locator Pactory - Singleton Pactory -
Active Record - Single Table Inheritance Adapter - Decorator Adapter - Decorator Adapter - Pacade Analogy Bridge - Strategy Analogy Builder - Decorator Analogy CQRS - Sente Sourcing CQRS - Sent Sour
Active Record - Single Table Inheritance Adapter - Decorator Adapter - Pacade Bridge - Strategy Analogy Builder - Decorator Analogy Builder - Decorator Analogy CQRS - Event Sourcing CQRS - Event Sourcing CQRS - Messaging Co-operation CQRS - Messaging Co-operation CQRS - Messaging Command - MVVM Dependency Command - Strategy Analogy
Adapter - Decorator Adapter - Beracde Analogy Bridge - Strategy Analogy Builder - Decorator Analogy CQRS - Event Sourcing CO-operation CQRS - Messaging Co-operation Command - MVVM Dependency DAO - Protory DAO - Protory DAO - Record Set DAO - Record Set DAO - Brotory DAO - Record Set DAO - Record Set DAO - Record Set DAO - Protory DAO - Record Set DAO - Rec
Analogy both of them decouple an abstraction from its implementation I builder - Decorator Analogy Builder - Decorator Analogy CQRS - Event Sourcing Co-operation CQRS - Event Sourcing CO-operation CQRS - Messaging Co-operation COmposite - Network Without Co-operation DAO - Ptro Co-operation DAO - Ptro Co-operation DAO - Record Set DAO - Pactory Co-operation DAO - Record Set DAO - Record Set DAO - Record Set DAO - Record Set DAO - Rependency DAO - Repository Data Mapper - Domain Model Co-operation DAO - Mescord Set DATE - Messaging Co-operation Co-operation DAO - Record Set DAO - Record
Bridge - Strategy
Builder - Decorator Builder - Pactory Analogy CQRS - Event Sourcing CORS - Co-operation CQRS - Messaging Co-operation CQRS - Messaging Command - MVVM Dependency Command - MVVM Dependency Composite - Iterator Composite - Iterator Composite - Visitor Composite - Visitor Composite - Visitor Composite - Visitor Composite - Repository DAO - Pactory DAO - Repository DAO - Repository DAO - MVVM Dependency DTO - Analogy DTO - MVVM DTO - Sepository DTO - Repository DTO - Repository DTO - Repository DTO - Sepository DEPendency Injection - Factory Dependency Injection - Factory Dependency Injection - Factory Dependency Injection - Strategy DEPENDENCE - Sepository Dependency Injection - Repository Dependency Injection - Service Locator Analogy Dependency I
Builder - Factory CQRS - Event Sourcing CC-operation CQRS is often used along with event sourcing for efficient queries messaging can be used to send commands to the domain in CQRS command - MVVM Dependency Listor Composite - Herator Composite - Visitor -
CQRS - Messaging Co-operation CQRS is often used along with event sourcing for efficient queries CQRS - Messaging Command - MVVM Dependency Command - MVVM Dependency Command - Strategy Analogy DAO - DTO DAO - DTO DAO - PTO DAO - PTO DAO - Repository DAO - Repository DTO - ANalogy DTO - AWC DTO - Sepository DEPendency Injection - Factory Dependency Injection - Factory Dependency Injection - Sepository Dependency Injection - Sepository Dependency Injection - Sepository Dependency Injection - Sepository Dependency Injection - Singleton Dependency Injection - Singleton Dependency Injection - Singleton Dependency Injection - Singleton Dependency Injection - Service Locator Dependency Injection - Singleton Dependency Injection - Service Locator Analogy Dependency Injection - Service Locator Dependency Injection - Servi
CQRS - Messaging Co-operation CQRS  Command - MVVM Dependency  Commosite - Iterator Co-operation DAO - DTO Analogy DAO - Record Set Dependency Data Mapper - Domain Model Co-operation DAO - Record Set Dependency Data Mapper - Domain Model Co-operation data Mapper - Domain Model Co-operation DAO - Record Set Dependency Data Mapper - Domain Model Co-operation data Mapper - Repository Da
CQRS - Messaging Command - MVVM Dependency Command - Strategy Analogy Composite - Iterator Composite - Iterator Composite - Visitor Co-operation Co-operation Composite - Visitor Interestor Visitor Interestor Co-operation Interestor Composite Interator Visitor Inte
Command - MVVM Dependency Command - Strategy Composite - Visitor to composite I the view with the viewmodel both patterns encapsulate an algorithm and decouple implementation details from their calling classes use iterator to recursive the structure of composite I the view with the viewmodel both patterns encapsulate an algorithm and decouple implementation details from their calling classes use iterator to recursive the structure of composite I patterns can be used in DAO to manipulate data I they are all responsible for data access of a software system I data mapper can be used for transferring data between the domain logic and the database I patterns can be used of Transferring data between the domain logic and the database I patterns can be used of MVVM I patterns can be used for MVVM I patterns can be used to add behaviours to the base composents I patterns can be used to add behaviours to the base composents I patterns can be used to add behaviours to the base composents I patterns can be used to add behaviours to the base composents I patterns can be used to add behaviours to the base composents I patterns can be used to add behaviours to the base composents I patterns can be used to add behaviours to the base composents I patterns can be used to add behaviours to the base composents I patterns can be used to add behaviours to the base composents I patterns can be used to add behaviours to the base compos
Command - MVVM Command - Strategy  Analogy Command - Strategy  Analogy Composite - Iterator Composite - Iterator Composite - Visitor Cooperation Composite - Visitor Cooperation Composite - Visitor Cooperation DAO - DTO Analogy DAO - Factory DAO - Factory DAO - Repository DAO -
Command - Strategy  Analogy  Composite - Iterator  Cooperation Composite - Visitor Cooperation DAO - DTO Analogy DAO - Record Set Dependency DAO - Record Set Data Mapper - Domain Model  Co-operation DTO - WC Dependency DTO - WVW DTO - Repository DTO - Service Layers DECORD - Interceptor DEPENDENCY - Analogy DTO - Service Layers DEPENDENCY - Analogy DECORATOR - Strategy DEPENDENCY - Analogy DEPNDENCY -
Composite - Iterator Co-operation DAO - DTO Analogy DAO - Factory Co-operation DAO - DTO Analogy Data Mapper - Repository Analogy DTO - MVC Dependency DTO - MVC Dependency DTO - Service Layers Dependency Injection - Factory Analogy Dependency Injection - Factory Analogy Dependency Injection - Reflection Dependency Injection - Service Locator - Repository Analogy Dependency Injection - Service Locator - Strategy Analogy Dependency Injection - Strategy Analogy Heave Injection - Heave I
Composite - Iterator Co-operation Composite - Visitor Co-operation DAO - DTO Analogy DAO - Record Set Dependency DAO - Record Set Dependency Data Mapper - Domain Model Co-operation Compositie of the Mark Mapper - Domain Model Co-operation Composite of Co-operation DAO - Record Set Dependency Data Mapper - Domain Model Co-operation Composite of Co-operation Composite Office of Set Dependency Data Mapper - Domain Model Co-operation Composite Co-operation Composite Office Offi
Composite - Iterator Cooperation Co-operation COoposite - Visitor
Co-operation DAO - DTO Analogy DAO - Becord Set Dependency DAO - Record Set Dependency DAO - Repository Data Mapper - Domain Model Co-operation DAO - Repository Data Mapper - Domain Model Co-operation DAO - Repository Data Mapper - Domain Model Co-operation DAO - Repository Data Mapper - Domain Model Co-operation data Mapper - Domain Model Co-operation DAO - Repository Data Mapper - Repository Dependency Data Mapper - Repository Data Mapper - Data Mapp
DAO - DTO DAO - Ractory DAO - Record Set DAO - Record Set DAO - Repository DAI Mapper - Domain Model DAI Mapper - Repository DAO - MVC Dependency DTO - MVC Dependency DTO - MVVM Dependency DTO - Repository DTO - Service Layers DTO - Service Layers DTO - Service Layers DTO - Service Layers DEPOSITION - Repository DEPOSITION - REPOSITION - Factory DEPOSITION - REPOSITORY
DAO - Factory Co-operation DAO - Record Set DAO - Repository Analogy Data Mapper - Domain Model Co-operation Data Mapper - Domain Model Co-operation DTO - MVVM Dependency DTO - Repository DTO - WVM Dependency DTO - Service Layers Dependency DTO - Strategy Analogy they are all responsible for creating its dependency injection - Repository Analogy Dependency DTO as the Model of MVV DTO - Analogy they are all used as data containers I DTO - Strategy Analogy they are all used as data containers I DTO - Strategy Analogy they are all used as data containers I Dependency Injection - Factory Analogy they are all used as data containers I Dependency Injection - Repository Co-operation Dependency Injection - Repository Co-operation Dependency Injection - Service Locator Analogy they are all used as data containers I Dependency Injection - Repository Co-operation Dependency Injection - Strategy Analogy they are all used as data containers I Dependency Injection - Repository Co-operation Dependency Injection - Service Locator Analogy they are all used as data containers I Dependency Injection - Repository Co-operation Dependency Injection - Service Locator Analogy they are all used as data containers I Dependency Injection - Repository Co-operation Dependency Injection - Singleton Analogy they are all used as data containers I Dotto and be decouple the ViewModel I I Dependency Injection - Repository Co-operation Dependency Injection - Service Locator Analogy they are all for data access of a software system I DTO - Analogy they are similar in implementations I DTO as the Model of MVC
DAO - Record Set DAO - Repository DAO - Repository DAO - Repository DAO - Repository DATE Analogy DATE ANALOG
DAO- Repository Data Mapper - Domain Model Co-operation Data Mapper - Repository Dependency Injection - Factory Data Mapper - Repository Dependency Injection - Factory Data Mapper - Repository Dependency Injection - Reflection Dependency Injection - Repository Dependency Injection - Repository Dependency Injection - Service Locator Data Mapper - Repository Data Mapper - Repository Data Mapper - Repository Data Mapper - Repository Data Malogy Data Maccess relevant patterns I Data Access relevant patterns I Date used in Survice Injection I Interceptor communication I Interceptor communication I Interceptor communication I Interceptor Communication I Interceptor Survice Injection - Factory Data Coroperation I Interceptor Classes may be targets of dependency injection I Interceptor Classes may be targets of dependency injection I Interceptor Classes may be targets of dependency injection on be used to decouple the ViewModel I Intervention of the Coroperation I Interceptor Classes may be targets of dependency injection on the used to decouple the ViewModel I Intervention of the
Data Mapper - Domain Model Co-operation Data Mapper - Repository Data Mapper - Repository Dro - MVC Dependency Dro - MVC Dependency Dro - MVVM Dependency Dro - Repository Analogy Dro - Repository Analogy Dro - Repository Analogy Dro - Service Layers Dependency Decorator - Interceptor Decorator - Strategy Dependency Injection - Reflection Dependency Injection - Reflection Dependency Injection - Reflection Dependency Injection - Service Locator Dependency Injection - Singleton Dependency Injection - Singleton Dependency Injection - Strategy Analogy Dependency Injection - Strategy Dependency Injection - Reflection Dependency Injection - Reflection Dependency Injection - Reflection Dependency Injection - Singleton Dependency Injection - Singleton Analogy Dependency Injection - Singleton Dependency Injection - Singleton Analogy Dependency Injection -
Data Mapper - Repository Data Mapper - Repository Data Mapper - Repository Dro - MVC Dependency Dro - MVC Dependency Dro - Stevice Layers Dependency Dro - Repository Dro - Service Layers Dependency Dro - Service Layers Dependency Dro - Value Object Analogy Dro - Value Object Analogy Dro - Value Object Analogy Dro - Strategy Analogy Dro - Strategy Dr
Data Mapper - Repository DTO - MVC Dependency DTO as the Model of MVC DTO - MVVM Dependency DTO as the Model of MVVM DTO - Repository DTO - Repository DTO - Repository DTO - Service Loavers Dependency DTO as the Model of MVVM IDO - Service Layers DEPENDENCY DTO - Service Locator DTO - Service Locator Packade - Mediator Factory - Strategy DTO as the Model of MVVM IDO - MVVM DTO - Repository DTO - Repository DTO - Service Locator DTO - Service Locator Packade - Mediator Factory - Strategy DTO can be used in service layer for communication DTO - Value Object Analogy both data access relevant patterns IDTO - Value Object Analogy both data access relevant patterns IDTO - Value Object Analogy both data access relevant patterns IDTO - Value Object Analogy both data access relevant patterns IDTO - Value Object Analogy both data access relevant patterns IDTO - Value Object Analogy both data access relevant patterns IDTO - Value Object Analogy both data access relevant patterns IDTO - Value Object Analogy both data access relevant patterns IDTO - Value Object or a be used in service layer for communication IDTO - Value Object Analogy both patterns can be used to add behaviours to the base component Interceptor - Service Locator Analogy Analogy Analogy Analogy Dependency Injection - Interceptor  Dependency Injection - Repository Dependency Injection - Repository Dependency Injection - Singleton Analogy
DTO - MVC DTO - MVVM Dependency DTO - Repository Analogy DTO - Repository Analogy DTO - Service Layers DEPORTOR - MVVM DEPORTOR - Service Layers DEPORTOR - Manalogy DTO - Value Object DEPORTOR - Manalogy DEPORTOR - Service Layers DEPORTOR - Manalogy DEPORTOR - MANAL
DTO - MVVM DTO - Repository Analogy DTO - Service Layers DTO - Service Layers DTO - Value Object Analogy DTO - Value Object Analogy DECORATOR - Interceptor DECORATOR - Strategy DEPENDENCY DEPENDENCY Analogy DEPENDENCY Analogy DEPENDENCY Analogy DEPENDENCY Analogy DEPENDENCY Analogy DEPENDENCY DEP
DTO - Repository DTO - Service Layers Dependency DTO Can be used in service layer for communication IDTO - Value Object Analogy Decorator - Interceptor Decorator - Strategy Analogy Dependency Injection - Factory Dependency Injection - Interceptor Dependency Injection - MVVM Dependency Injection - Reflection Dependency Injection - Repository Dependency Injection - Service Locator Dependency Injection - Service Locator Dependency Injection - Singleton Dependency Injection - Service Locator Dependency Injection - Strategy Dependency Injection - Service Locator Dependency Injection - Service Locator Dependency Injection - Strategy Dependency Injection - Service Locator Dependency Injection - Strategy Double Checked Locking - Singleton Analogy Analogy Dependency Injection - Strategy Double Checked Locking - Singleton Analogy Analogy Dependency Injection Analogy Dependency Injection - Strategy Double Checked Locking - Singleton Analogy Analogy Dependency Injection Analogy Dependency Injection - Strategy Double Checked Locking - Singleton Analogy Analogy Dependency Injection Analogy Dependency Injection - Strategy Double Checked Locking - Singleton Analogy Dependency Injection - Strategy Double Checked Locking - Singleton Analogy Dependency Injection - Strategy Dependency Injection can be used to decouple the ViewModel Injection Interceptor Injection Interceptor Interceptor Interceptor Injection Interceptor Injection Interceptor Injection Interceptor Interceptor Injection Interceptor Injection Interceptor Interceptor Interceptor Injection Interceptor Interceptor Injection Interc
DTO - Service Layers DTO - Value Object Analogy DEcorator - Interceptor Analogy Decorator - Strategy Analogy Dependency Injection - Factory Dependency Injection - MVVM Dependency Injection - Reflection Dependency Injection - Repository Dependency Injection - Service Locator Dependency Injection - Service Locator Dependency Injection - Singleton Dependency Injection - Singleton Analogy Dependency Injection - Strategy Dependency Injection - Singleton Analogy Dependency Injection - Singleton Analogy Dependency Injection - Strategy Analogy Analo
DTO - Value Object Analogy they are all used as data containers I Decorator - Interceptor Analogy they achieve similar functions I Decorator - Strategy Analogy both patterns can be used to add behaviours to the base components  Dependency Injection - Factory Analogy they all have the purpose to separate the use of a certain of component interceptor classes may be targets of dependency injection, I e.g., in Context and Dependency Injection (CDI) dependency Injection - MVVM  Dependency Injection - Reflection Dependency dependency injection can be used to decouple the ViewModel in MVVM  Dependency Injection - Repository Co-operation (Dependency Injection - Service Locator Analogy they are all used as data containers  I Dependency Injection - Factory Analogy they can all make dependency injection (CDI) dependency Injection can be used to decouple the ViewModel in MVVM  Dependency Injection - Repository Co-operation the class is still responsible for creating its dependencies in both patterns  Dependency Injection - Singleton Analogy they can all make dependencies for objects I Dependency Injection - Strategy Analogy they all allow us to set run-time behaviours of objects I Safe  Facade - Mediator Analogy they are simplified version of factory method I Factory - Service Locator Analogy both encapsulate the functionalities of systems I Factory - Service Locator Analogy they are all for creating objects I Factory - Strategy Analogy they are all for creating objects I HWC is a variation of MVC I HWC Analogy HWC is a variation of MVC I I HWC is a variation of MVC I I HWC is a variation of MVC I I I HWC is a variation of which is tructures of I I I I I I I I I I I I I I I I I I
Decorator - Interceptor Decorator - Strategy Analogy both patterns can be used to add behaviours to the base components  Dependency Injection - Factory Analogy they all have the purpose to separate the use of a certain component interceptor classes may be targets of dependency injection, e.g., in Context and Dependency Injection (CDI) dependency Injection - Reflection Dependency Injection can be used to decouple the ViewModel in MVVM  Dependency Injection - Reflection Dependency dependency injection can be used to decouple the ViewModel in MVVM  Dependency Injection - Repository Co-operation Pependency Injection can be implemented by using reflection repositories can be injected via dependency injection Interceptor classes in a both patterns  Dependency Injection - Service Locator Analogy they can all make dependencies for objects Industry analogy they all allow us to set run-time behaviours of objects Industry and objects Industry Analogy they all encapsulate the functionalities of systems Infactory Method - Factory Variation factory is a simplified version of factory method Infactory - Service Locator Analogy they are similar in implementations Infactory Injection in Infactory Injection Infactory Injectio
Dependency Injection - Factory Dependency Injection - MVVM Dependency Injection - Reflection Dependency Injection - Reflection Dependency Injection - Reflection Dependency Injection - Reflection Dependency Injection - Respository Dependency Injection - Service Locator Dependency Injection - Singleton Dependency Injection - Singleton Dependency Injection - Strategy Dependency Injection - Strategy Dependency Injection - Strategy Double Checked Locking - Singleton Factory - Service Locator Analogy Teactory - Service Locator Factory - Service Locator Analogy Dependency - Service Locator Dependency - Singleton Dependency - Singleto
Dependency Injection - Factory  Analogy  Dependency Injection - Interceptor  Dependency Injection - Interceptor  Dependency Injection - MVVM  Dependency Injection - Reflection  Dependency Injection - Reflection  Dependency Injection - Repository  Dependency Injection - Service Locator  Dependency Injection - Singleton  Analogy  Analogy  They and Imake dependencies for objects  Dependency Injection - Singleton  Dependency Injection - Singleton  Analogy  They and Imake dependencies for objects  I Dependency Injection - Singleton  Analogy  They and Imake dependencies for objects  I Double Checked Locking - Singleton  Dependency  Depend
Dependency Injection - Factory  Dependency Injection - Interceptor  Co-operation  Dependency Injection - MVVM  Dependency Injection - Reflection  Dependency Injection - Reflection  Dependency Injection - Reflection  Dependency Injection - Reflection  Dependency Injection - Service Locator  Dependency Injection - Singleton  Dependency Injection - Singleton  Dependency Injection - Strategy  Analogy  They all allow us to set run-time behaviours of objects  Injection - Strategy  How Injection - Interceptor Injection
Dependency Injection - Interceptor  Co-operation interceptor classes may be targets of dependency injection, e.g., in Context and Dependency Injection (CDI)  Dependency Injection - MVVM  Dependency Injection - Reflection Dependency Injection - Reflection Dependency Injection - Repository Dependency Injection - Repository Dependency Injection - Service Locator  Analogy Dependency Injection - Singleton Dependency Injection - Strategy Double Checked Locking - Singleton Facade - Mediator Facade - Mediator Factory - Service Locator Analogy Analogy Dependency Injection - Singleton Factory - Service Locator Analogy Service Locator Analogy HMVC - MVC Analogy HMVC - MVC Analogy HMVC is a variation of MVC I Interator - Visitor  I context and Dependency Injection (CDI) Dependency Injection can be used to decouple the ViewModel I in MVVM  dependency Injection can be insplemented by using reflection I repositories can be injected via dependency injection I repositories can be injected via dependency injection I repositories can be injected via dependency injection I both patterns  be in MVVM  The class is still responsible for creating its dependencies in both patterns  I both patterns  Lepandency Injection - Strategy Analogy They an all make dependencies for objects I double checked locking can be used to make singleton thread-safe  safe  Safe  Gueva Analogy They all allow us to set run-time behaviours of objects I double checked locking can be used to make singleton thread-safe  safe  Safe  Factory - Singleton  Factory - Service Locator Analogy They all encapsulate the functionalities of systems I service - Service Locator Analogy They all encapsulate the creation of factory method I service - Service Locator Factory - Singleton Factory - Singleton Factory - Singleton Factory - Singleton Factory
Dependency Injection - Interceptor  Dependency Injection - MVVM  Dependency Injection - Reflection  Dependency Injection - Reflection  Dependency Injection - Reflection  Dependency Injection - Reflection  Dependency Injection - Repository  Dependency Injection - Service Locator  Dependency Injection - Singleton  Analogy  Dependency Injection - Strategy  Double Checked Locking - Singleton  Factory Method - Factory  Factory - Service Locator  Analogy  Analogy  The class is still responsible for creating its dependencies in both patterns  Dependency Injection - Strategy  Dependency Injection - Strategy  Analogy  The class is still responsible for creating its dependencies in both patterns  Dependency Injection - Strategy  Dependency Injection - Strategy  Analogy  The class is still responsible for creating its dependencies in both patterns  In the class is still responsible for creating its dependencies in both patterns  In the class is still responsible for creating its dependencies in both patterns  Dependency Injection - Strategy  Analogy  The class is still responsible for creating its dependencies in both patterns  In the class is still responsible for creating its dependencies in both patterns  In the class is still responsible for creating objects  In the class is still responsible for creating objects  In the class is still responsible for creating objects  In the class is still responsible for creating objects  In the class is still responsible for creating objects  In the class is still responsible for creating objects  In the class is still responsible for creating objects  In the class is still responsible for creating objects  In the class is still responsible for creating objects  In the class is still responsible for creating objects  In the class is still responsible for creating objects  In the class is still responsible for creating objects  In the class is still responsible for creating objects  In the class is still responsible for creating objects  In the class is still responsible for creatin
Dependency Injection - MVVM  Dependency Injection - Reflection  Dependency Injection - Repository  Dependency Injection - Service Locator  Dependency Injection - Service Locator  Dependency Injection - Singleton  Analogy  Analogy  Analogy  Analogy  They can all make dependencies for objects  Dependency Injection - Strategy  Double Checked Locking - Singleton  Dependency  Analogy  They all allow us to set run-time behaviours of objects  Jouble Checked Locking - Singleton  Dependency  They all encapsulate the functionalities of systems  Infectory - Service Locator  Factory - Service Locator  Analogy  They are all for creating objects  Infectory - Singleton  Analogy  They are all for creating objects  Infectory - Strategy  Analogy  They are all for creating objects  Infectory - Strategy  Analogy  They are all for creating objects  Infectory - Strategy  Analogy  They are all for creating objects  Infectory - Strategy  Analogy  They are all for creating objects  Infectory - Strategy  Analogy  They are similar in implementations  Infectory - MVC  Analogy  They are similar in implementations  Infectory - MVC  Analogy  They are similar in implementations  Infectory - MVC  Analogy  They are similar in a visitor can be used to visit structures of Infectory - Visitor  The treator - Visitor  The treator - Visitor - Analogy  They are similar in visitor can be used to visit structures of Infectory - Visitor  The treator - Visitor - Analogy  They are all for creating objects  The treator - Visitor - Analogy  They are similar in implementations  The treator - Visitor - Analogy  They are similar in visitor can be used to visit structures of Infectory  The treator - Visitor - Analogy  The treator
Dependency Injection - MVVM  Dependency Injection - Reflection  Dependency Injection - Reflection  Dependency Injection - Reflection  Dependency Injection - Repository  Dependency Injection - Service Locator  Dependency Injection - Service Locator  Dependency Injection - Singleton  Analogy  They can all make dependencies for objects  Dependency Injection - Strategy  Dependency Injection - Strategy  Analogy  They all allow us to set run-time behaviours of objects  I Double Checked Locking - Singleton  Dependency  Facade - Mediator  Factory Method - Factory  Factory - Service Locator  Factory - Service Locator  Factory - Singleton  Analogy  They all encapsulate the functionalities of systems  I Factory - Service Locator  Factory - Singleton  Analogy  They are all for creating objects  I Factory - Strategy  Analogy  They are all for creating objects  I HMVC - MVC  Analogy  HMVC is a variation of MVC  I Interator - Visitor
Dependency Injection - Reflection Dependency Injection - Repository Dependency Injection - Service Locator Dependency Injection - Service Locator Dependency Injection - Singleton Dependency Injection - Strategy Dependency Injection in Injection Inj
Dependency Injection - Reflection Dependency Injection - Repository Dependency Injection - Service Locator Dependency Injection - Service Locator Dependency Injection - Singleton Dependency Injection - Strategy Analogy Double Checked Locking - Singleton Dependency
Dependency Injection - Repository Dependency Injection - Service Locator Dependency Injection - Service Locator Dependency Injection - Singleton Dependency Injection - Singleton Dependency Injection - Singleton Dependency Injection - Strategy Analogy Double Checked Locking - Singleton Dependency Dependencies for objects I Dependencies in I Dependencies in I Dependencies for objects I Dependencies in I Dependencies in I Dependencies in I Dependency I analogy I they all allow us to set run-time behaviours of objects I Dependencies in I Dependency I analogy I they all encapsulate the functionalities of systems I factory is a simplified version of factory method I Factory - Service Locator Analogy I they are all for creating objects I Factory - Strategy Analogy I they are similar in implementations I HMVC - MVC Analogy I HMVC is a variation of MVC I I Iterator - Visitor I both patterns I the class is still responsible for creating its dependencies in I the class is still responsible for creating its dependencies in I the class is still responsible for creating its dependencies in I both patterns I they all encapsulate the functionalities of systems I factory is a simplified version of factory method I factory is a simplified version of factory method I factory is a simplified version of factory method I factory is a simplified version of factory method I factory is a simplified version of factory method I factory is a simplified version of factory method I factory is a simplified version of factory method
Dependency Injection - Service Locator  Dependency Injection - Singleton  Analogy  Analogy  They can all make dependencies for objects  I Dependency Injection - Strategy  Analogy  Analogy  They all allow us to set run-time behaviours of objects  I Double Checked Locking - Singleton  Dependency  Dependency  Dependency  Dependency  They all allow us to set run-time behaviours of objects  I Double Checked Locking - Singleton  Dependency  They all encapsulate the functionalities of systems  I Facade - Mediator  Factory Method - Factory  Variation  Factory - Service Locator  Analogy  They are all for creating objects  I Factory - Strategy  Analogy  They are similar in implementations  I HMVC - MVC  Analogy  HMVC is a variation of MVC  I I Stratory - Visitor  I Service Locator is a simplified version of the objects  I HMVC is a variation of MVC  I I Stratory - Visitor  I Service Locator is a simplified version of the objects  I Service Locator  Analogy  They are similar in implementations  I HMVC is a variation of MVC  I I Service Locator is a variation of MVC  I Service Locator is a variatio
both patterns  Dependency Injection - Singleton Analogy they can all make dependencies for objects I  Dependency Injection - Strategy Analogy they all allow us to set run-time behaviours of objects I  Double Checked Locking - Singleton Dependency double checked locking can be used to make singleton threadsafe  Facade - Mediator Analogy they all encapsulate the functionalities of systems I  Factory Method - Factory Variation factory is a simplified version of factory method I  Factory - Service Locator Analogy both encapsulate the creation of the objects I  Factory - Strategy Analogy they are all for creating objects I  Factory - Strategy Analogy they are similar in implementations I  HMVC - MVC Analogy Both iterator and visitor can be used to visit structures of I
Dependency Injection - Singleton Dependency Injection - Strategy Analogy Double Checked Locking - Singleton Dependency Dependency Double Checked locking can be used to make singleton thread-safe  Facade - Mediator Factory Method - Factory Variation Factory is a simplified version of factory method I Factory - Service Locator Analogy Double Checked locking can be used to make singleton thread-safe I Factory Method - Factory Variation Factory is a simplified version of factory method I Factory - Service Locator Analogy Doth encapsulate the creation of the objects I Factory - Strategy Analogy HMVC is a variation of MVC I Iterator - Visitor Analogy Doth iterator and visitor can be used to visit structures of
Dependency Injection - Strategy Double Checked Locking - Singleton Dependency
Double Checked Locking - Singleton  Dependency Safe  Facade - Mediator Factory Method - Factory Variation Factory - Service Locator Factory - Singleton Analogy Analogy They all encapsulate the functionalities of systems I Factory is a simplified version of factory method I Factory - Service Locator Analogy Both encapsulate the creation of the objects I Factory - Singleton Factory - Strategy Analogy HMVC - MVC Analogy HMVC is a variation of MVC I Iterator - Visitor  Dependency double checked locking can be used to make singleton thread- safe  I  HMVC - MVC Analogy HMVC is a variation of MVC I I I terator - Visitor  I ouble checked locking can be used to make singleton thread- safe  I  L  L  L  L  L  L  L  L  L  L  L  L
Safe  Facade - Mediator  Factory Method - Factory  Variation  Factory - Service Locator  Factory - Singleton  Factory - Strategy  Analogy  Analogy  Analogy  They are all for creating objects  Factory - Strategy  Analogy  HMVC - MVC  Analogy  HMVC is a variation of MVC  Iterator - Visitor  Safe  safe  safe  they all encapsulate the functionalities of systems  I  Factory is a simplified version of factory method  I  Factory is a simplified version of the objects  I  Factory - Singleton  Analogy  Hey are all for creating objects  I  HMVC is a variation of MVC  I  Iterator - Visitor  I  Safe  Safe  Analogy  I  I  I  I  I  I  I  I  I  I  I  I  I
Facade - Mediator Analogy Variation Factory Method - Factory Variation Factory is a simplified version of factory method I Factory - Service Locator Analogy both encapsulate the creation of the objects I Factory - Singleton Analogy they are all for creating objects I Factory - Strategy Analogy they are similar in implementations I HMVC - MVC Analogy HMVC is a variation of MVC I Iterator - Visitor Analogy both iterator and visitor can be used to visit structures of
Factory Method - Factory  Factory Method - Factory  Factory - Service Locator  Factory - Service Locator  Factory - Singleton  Analogy  Analogy  They are all for creating objects  Factory - Strategy  Analogy  They are similar in implementations  I  HMVC - MVC  Analogy  HMVC is a variation of MVC  Iterator - Visitor  Analogy  They are similar in implementations  I  HMVC is a variation of MVC  I  South iterator and visitor can be used to visit structures of  I
Factory - Service Locator Analogy both encapsulate the creation of the objects I Factory - Singleton Analogy they are all for creating objects I Factory - Strategy Analogy they are similar in implementations I HMVC - MVC Analogy HMVC is a variation of MVC I Iterator - Visitor Analogy both iterator and visitor can be used to visit structures of I
Factory - Singleton Analogy they are all for creating objects I Factory - Strategy Analogy they are similar in implementations I HMVC - MVC Analogy HMVC is a variation of MVC I Iterator - Visitor Analogy both iterator and visitor can be used to visit structures of I
Factory - Strategy Analogy they are similar in implementations I HMVC - MVC Analogy HMVC is a variation of MVC I Iterator - Visitor Analogy both iterator and visitor can be used to visit structures of I
HMVC - MVC Analogy HMVC is a variation of MVC I I Iterator - Visitor Analogy both iterator and visitor can be used to visit structures of I
Iterator - Visitor Analogy both iterator and visitor can be used to visit structures of I
elements
Lazy Loading - Singleton Dependency lazy loading can be used to make singleton thread-safe I
Master/Slave - Sharding Analogy they can be all database partitioning approaches I
Mediator - MVVM Dependency use mediator to implement communication between View- I
Models in MVVM
Mediator - Observer Analogy they have similar functions I
Mediator - Publish/Subscribe Dependency mediator can be used to implement the publish/subscribe I
model
Message Broker - Messaging Variation message broker is a way for messaging I
Message Broker - Publish/Subscribe Dependency use message broker to implement publish-subscribe messaging I
Messaging - Publish/Subscribe Variation publish/subscribe is a kind of messaging pattern I
MVC - MVP Analogy MVC and MVP are all for building architectures of presenta-
tion NUMBER OF THE PROPERTY OF
MVC - MVVM Analogy MVC and MVVM are all for building architectures of presen-
tation  NVC Observer and the West and the West and the Wiser I
MVC - Observer Dependency observer can be used to synchronize the Model and the View I
in MVC  MVC Peneritary:  Co-eneration the repositories interest with the Controller in MVC
MVC - Repository Co-operation the repositories interact with the Controller in MVC I  MVC - Service Layers Co-operation service layers interact with the controller in MVC I
in to better Edycis competation better tayons invertee with the constraint in in the
MVP - MVVM Analogy MVP and MVVM are all for building architectures of presen-
tation  MYVM Repository Co-providen repository can be used with the Medal in MYVM
MVVM - Repository Co-operation repository can be used with the Model in MVVM I
Observer - Publish/Subscribe Variation publish/subscribe pattern is a variation of observer pattern I
Reflection - Singleton Co-operation singletons should be made reflection-proof
Repository - Service Layers Co-operation repository can be used in the data access layer to accompany I
with the service layer  Repository - Unit of Work  Co-operation  with the service layer  unit of work is often implemented on repositories  I
State - Strategy Analogy the two patterns are pretty similar in practice I Strategy - Visitor Analogy they have similar behaviours I