availability

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| pattern | Availability | | | | | | | | | | | | |
| DetectFaults | | | Recovery - Preparation | | | | Recovery - Reintroduction | | Prevention |  |  | | |
| Ping/Echo | Heartbeat | Exception | Active Redundancy | Passive Redundancy | Spare | Shadow | | StateResynchronization | Removal from Service | Transcations | Process Monitor |
| Layered |  |  | x |  |  |  |  | |  |  |  |  |
| Pipes and Filters | x |  | x |  |  |  |  | |  |  | x |  |
| Blackboard |  |  |  |  | x |  |  | | x |  |  |  |
| Broker | x |  | x | x |  |  |  | | x |  |  |  |
| MVC |  |  | x |  |  |  |  | |  |  |  |  |
| C/S | x |  | x |  |  |  |  | | x |  |  |  |
| P2P | x |  | x |  |  |  |  | |  |  |  |  |
| SOA |  | x | x |  |  |  |  | |  |  |  |  |
| Publish-Subscribe |  |  | x |  |  |  |  | | x |  |  |  |
| Presentation Abstraction Control |  |  | x |  |  |  |  | |  |  |  |  |

**benefits**：

Pipes and Filters：Has a means to detect and handle faults,so that it can be detected and responded in time

Broker：Has a means to detect and handle faults,so that it can be detected and responded in time

P2P：Multiple paths are available.

SOA：Provide a reliable mechanism to support

**penalties**:

When the complexity of each pattern increases, it will cause certain instability.

**not associated：**

Shadow、ProcessMonitor：Related to hardware, there are not many applications in software

interoperability

|  |  |  |  |
| --- | --- | --- | --- |
| pattern | Interoperability | | |
| Locate | Manage Interfaces | |
| Discover Service | Orchestrate | Tailor Interface |
| Layered |  | x | x |
| Pipes and Filters |  | x | x |
| Blackboard |  | x | x |
| Broker | x | x | x |
| MVC |  | x | x |
| C/S |  | x | x |
| P2P |  | x | x |
| SOA | x | x | x |
| Publish-Subscribe |  | x | x |
| Presentation Abstraction Control |  | x | x |

**benefits**：

These patterns have good interface design and encapsulation, so they have high interoperability.

performance

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| pattern | Performance | | | | | | |
| Control Resource Demand | | | | Manage Resources | | |
| Increase Computation Efficiency | Limit Event Response | Reduce Overhead | Bound Execution Times | Introduce  Concurrency | Increase Resources | Schedule Resources |
| Layered |  |  |  |  |  |  |  |
| Pipes and Filters |  |  |  |  | x |  |  |
| Blackboard |  |  |  |  |  |  |  |
| Broker |  |  |  |  |  |  |  |
| MVC |  |  |  |  |  |  |  |
| C/S | x |  |  |  |  |  |  |
| P2P | x |  |  |  |  |  | x |
| SOA |  |  |  |  |  |  |  |
| Publish-Subscribe |  |  |  |  | x |  |  |
| Presentation Abstraction Control |  |  |  |  | x |  |  |

**benefits**：

Pipes and Filters：Improve performance.Introducing concurrency.

P2P：In some cases, P2P systems can achieve direct data transfer, thereby improving performance.

Publich-Subscribe：Takes a parallel mechanism

PAC：Completely isolate presentation and abstraction based on mvc and improve the performance.

**penalties**:

Layered：Reduce the performance.

Pipes and Filters：Performance is affected when dealing with interactions

Blackboard：Increased complexity leads to reduced efficiency

Broker：Reduce some performance. Added a layer of Broker message forwarding, which has reduced efficiency.

MVC：Depending on the model's operational interface, the view may need to be called multiple times to get enough display data. Unnecessary frequent access to unchanging data will also impair operational performance.

P2P：Small P2P systems may not to consistently achieve quality goals such as perfomance and availability.

PAC：Depending on the model's operational interface, the view may need to be called multiple times to get enough display data. Unnecessary frequent access to unchanging data will also impair operational performance.

**not associated：**

Increase Resources：This tactics can be placed in the deployment of hardware resources, which is not obvious in the design pattern.

security

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| pattern | Security | | | | | | | | |
| Detect Attacks | | Resist Attacks | | | React to Attacks | | Recover from Attacks | |
| Detect Intrusion | Message Integrity | Identify Actors | Encrypt Data | Separate Entities | Revoke Access | Lock Computer | Maintain Audit Trail | Restore |
| Layered |  |  |  |  |  |  |  |  |  |
| Pipes and Filters |  |  | x | x |  |  | x |  |  |
| Blackboard |  |  |  |  |  |  |  |  |  |
| Broker |  |  | x | x |  | x | x |  | x |
| MVC |  |  |  |  |  |  |  |  |  |
| C/S | x |  | x |  |  | x |  | x | x |
| P2P |  |  |  |  |  |  |  |  |  |
| SOA | x |  | x | x |  | x |  | x | x |
| Publish-Subscribe |  |  |  | x |  | x |  | x |  |
| Presentation Abstraction Control |  |  |  |  |  |  |  |  |  |

**benefits**：

Pipes and Filters：Provide the ability to detect intrusions and encrypts information, improving system security.

Broker：Provide the ability to detect intrusions and encrypts information, improving system security.

C/S：Improve security. All data is stored on the server, and the server has better control over access

SOA：More robust security mechanism

**penalties**:

P2P：Reduce security. P2P system cannot identify user identity due to environmental restrictions

**not associated：**

Separate Entities： based on hardware

testability

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| pattern | Testability | | | | |
| Control and Observe System State | | | Limit Complexity | |
| Specialized Interfaces | Record/Playback | Abstract Data Sources | Limit Structural Complexity | Limit Nondeterminism |
| Layered | x |  | x | x | x |
| Pipes and Filters |  |  | x | x | x |
| Blackboard |  |  | x |  |  |
| Broker |  |  |  | x |  |
| MVC | x |  | x |  |  |
| C/S | x |  |  | x |  |
| P2P |  |  |  |  |  |
| SOA | x |  |  |  |  |
| Publish-Subscribe | x |  |  | x |  |
| Presentation Abstraction Control | x |  | x |  |  |

**benefits**：

Layered：Testability can be enhanced because each layer can be tested independently.

Pipes and Filters：Divide functions and control complexity to a certain extent

MVC：Testing can be performed separately in m, v, and c

C/S：The client and the server can be tested separately and then tested interactively.Improve the **testability**.

Publish-Subscribe

PAC

**penalties**:

Blackboard：Reduce testability(Increased complexity)

Broker：Testability requires attention with increased testing work.

**not assosiate:**

executable assertions、sandbox：is a kind of technical means

usability

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| pattern | Usability | | | | | | |
| Support User Initiative | | | | Support System Initiative | | |
| Cancel | Undo | Pause/Resume | Aggregate | Maintain Task Model | Maintain User Model | Maintain System Model |
| Layered |  |  |  |  |  | x |  |
| Pipes and Filters | x | x | x |  | x |  | x |
| Blackboard |  |  |  |  | x | x | x |
| Broker | x | x | x |  |  |  |  |
| MVC |  |  |  |  |  | x |  |
| C/S | x | x | x |  | x | x | x |
| P2P | x | x | x |  | x | x | x |
| SOA | x | x | x |  | x | x | x |
| Publish-Subscribe | x | x | x |  | x | x | x |
| Presentation Abstraction Control |  |  |  |  |  | x |  |

**benefits**：

Pipes and Filters：Communication detection can be performed while connected

SOA：Due to the loose coupling between service providers and service users, and the adoption of open standard interfaces, it has maintainability and usability.

C/S

P2P

Publish-Subscribe

**penalties**:

Layered：Reduce the usability.

**not assosiated：**

Aggregation cannot be implemented in architecture design