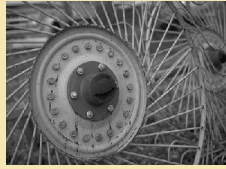


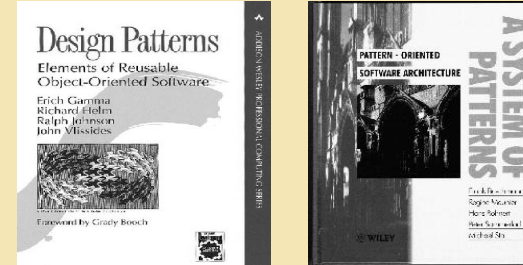
Patterns



Olga Danylevych
Universität Stuttgart Fakultät Informatik

Inhalt

- Die Herkunft des Begriffes
- Aspekte: Klassifikationsschemen und Beschreibung, Beispiel (Proxy)
- Einsatzbereiche von Mustern
- Ausblick: Schwierigkeiten und Vorteile



Erich Gamma, Richard Helm,
Ralph Johnson and John Vlissides
Addison Wesley, October
1994.
ISBN 0-201-63361-2

F. Buschmann, R. Meunier, H. Rohnert
P.Sommerlad, M. Stal
John Wiley and Sons Ltd, Chichester, UK,
1996
ISBN 0-471-95869-7

Der Begriff

Ein Muster...

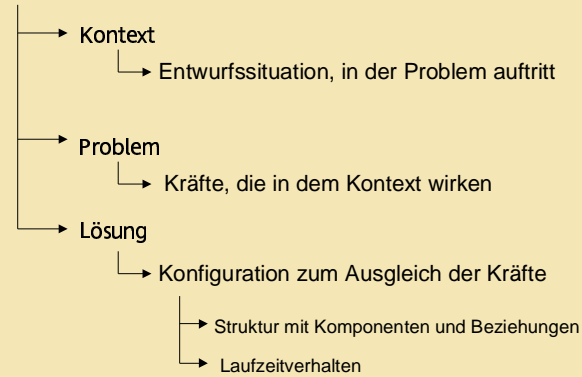
- löst ein Problem
- bietet ein erprobtes Konzept
- Lösung nicht offensichtlich
- beschreibt eine Beziehung
- hilft Menschen



PATTERN

„Ein Muster ist eine dreiteilige Regel, die eine Beziehung zwischen einem bestimmten Kontext, einem Problem und einer Lösung beschreibt.“ (Ch. Alexander)

MUSTER



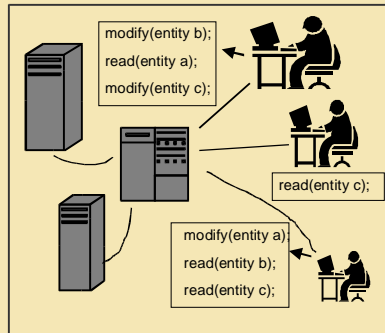
PROXY-Muster führt einen Stellvertreter für eine Komponente ein

Anforderungen

•Erhöhung der Effizienz

•Vereinfachung des Zugriffes

•Schutz von unberechtigtem Zugriff



30.10.02

Ferienakademie 2002

5

Klassifikationsschemen

1. Klassifikation von Muster nach GoV

(F. Buschmann, R. Meunier, H. Rohnert
P.Sommerlad, M. Stal)

1.1 Musterkategorien

1.2 Problemkategorien (im Bezug auf Entwurfsmuster)

6. Klassifikation von Entwurfsmuster nach GoF

(Erich Gamma, Richard Helm,
Ralph Johnson and John Vlissides)

30.10.02

Ferienakademie 2002

6

1.1 Musterkategorien (Architekturmustern)

Architekturmuster „spiegelt ein grundsätzliches Strukturierungsprinzip von Software-Systemen wider“

- beschreibt Menge vordefinierter Subsysteme
- spezifiziert deren jeweiligen Zuständigkeitsbereich
- enthält Regeln zur Organisation der Beziehungen zwischen den Subsystemen

(z.B. Model-View-Controller)

30.10.02

Ferienakademie 2002

7

1.1 Musterkategorien (Idiomen)

Idiom ist ein für eine bestimmte Programmiersprache spezifisches Muster auf einer niedriger Anstraktionsebene.

- beschreibt wie man spezielle Aspekte von Komponenten oder den Beziehungen zwischen ihnen mit den Mitteln der Programmiersprache implementieren kann.

(z.B. Singleton-Muster in C++ oder Smalltalk)

30.10.02

Ferienakademie 2002

8

1.1 Musterkategorien (Entwurfsmustern)

Entwurfsmuster „beschreibt ein Schema zur Verfeinerung von Subsystemen oder Komponenten eines Software -Systems oder Beziehungen zwischen ihnen.“

- Es beschreibt eine häufig auftretende Struktur von miteinander kommunizierenden Komponenten, die ein allgemeines Entwurfsproblem in einem Kontext löst

(z.B. Master-Slave, Observer, Proxy)

1.2 Problemkategorien von Entwurfsmustern (nach GoV)

- **Strukturelle Zerlegung** (z.B. Whole - Part)
- **Organisation der Arbeit** (z.B. Master - Slave)
- **Zugriffskontrolle** (z.B. Proxy)
- **Management** (z.B. Command-Processor)
- **Kommunikation** (z.B. Forwarder-Receiver)

2. Klassifikation von Entwurfsmustern (nach GoF)

G ü t i g k e i t s b e r e i c h	Aufgabe			
	Klassen- basiert	Erzeugungsmuster	Strukturmuster	Verhaltensmuster
		Fabrikmethode	Adapter (klassenbasiert)	Interpreter, Schablonen- methode
	Objekt- basiert	Abstrakte Fabrik Erbauer Prototyp Singleton	Adapter (objekt- basiert) Brücke Dekorierer Fassade Fliegengewicht Kompositum Proxy	Befehl Beobachter Besucher, Iterator Memento Strategie Vermittler, Zustand Zuständigkeits- kette

Beschreibung von Mustern

- | | |
|---------------------|--------------------|
| •Name | •Implementierungen |
| •Auch bekannt als | •Musterlösungen |
| •Beispiel | •Varianten |
| •Kontext | •Anwendungen |
| •Problem | •Auswirkungen |
| •Lösung | •Verweise |
| •Struktur | |
| •Dynamische Aspekte | |

PROXY-Muster führt einen Stellvertreter für eine Komponente ein

- Erhöhung der Effizienz
- Vereinfachung des Zugriffs
- Schutz von unberechtigtem Zugriff

The diagram illustrates the Proxy pattern. A client (represented by a person at a computer) sends requests to a proxy (represented by a server rack). The proxy then forwards these requests to one or more real objects (represented by server racks). The proxy acts as an intermediary, managing access to the real objects. The real objects perform the actual operations, such as reading or modifying data.

30.10.02 Ferienakademie 2002 13

PROXY-Muster (Struktur)

Abstrakte Basisklasse

The diagram shows the structural aspect of the Proxy pattern. It features an abstract base class (Abstrakte Basisklasse) at the top. Below it are two concrete classes: 'Original' and 'Stellvertreter'. Both 'Original' and 'Stellvertreter' inherit from the abstract base class, as indicated by the red arrows pointing upwards.

30.10.02 Ferienakademie 2002 14

PROXY-Muster DYNAMISCHE ASPEKTE

Aktivitäten des Stellvertreters – unabhängig von seiner Spezialisierung (Varianten)

- Vorverarbeitung: z.B. Nachschlagen der Adresse des Originals, Überprüfung, ob die Anforderte Information im lokalen Zwischenspeicher liegt
- Weiterleiten der Anforderung: Kommunikationsprotokoll Sicherheitsvorkehrungen
- Nachverarbeitung: z.B. Antwort im Zwischenspeicher ablegen, Destruktor des Originals aufrufen

The sequence diagram shows the interaction between the Client, Proxy, and RealSubject. The Client sends a 'doTask()' message to the Proxy. The Proxy then sends a 'service()' message to the RealSubject. The RealSubject performs the task and returns the result to the Proxy. The Proxy then performs 'pre-processing()' and 'post-processing()' before returning the result to the Client.

30.10.02 Ferienakademie 2002 15

Proxy-Muster Implementierung

- Identifizierung aller Verantwortungsbereiche
- Einführung der abstrakten Basisklasse -> Adapter-Muster
- Implementierung der Funktionalität des Stellvertreters
- Befreiung des Originals und seiner Klienten von den Verantwortungsbereichen, die dem Proxy übertragen werden
- Den Proxy mit dem Original in Verbindung bringen
- Jede direkte Verbindung zwischen dem Original und seinen Klienten ersetzen

30.10.02 Ferienakademie 2002 16

Varianten von Proxy

- Remote-Proxy
- Protection-Proxy
- Cache-Proxy
- Synchronisations-Proxy
- Counting-Proxy
- Virtual-Proxy
- Firewall-Proxy

30.10.02

Ferienakademie 2002

17

Muster in Software-Architektur

- Reihe von Mustern für Programmierung von Bedienoberflächen, verteilte Anwendungen sind schon vorhanden
- „Weise Flecken“ : Sicherungssysteme, Scientific Computing
Transaktions-Überwachungs-Systeme,
Parallelität, fehlertolerante Systeme
- Implementierung von Idiomen in wichtigsten Programmiersprachen
- Herstellung von Sammlungen von Mustern (für Verständnis & Anwendung der Bibliotheken, Frameworks und „middleware“ Plattformen)

30.10.02

Ferienakademie 2002

18

Einsatz(2)

Organisationsmuster

- Unterstützung von Management von Software-Entwicklungsprojekten
- Organisation der Anforderungsanalyse...

Mustersprachen

- Architektur (Christopher Alexander)
- CHECKS (Integritätsregeln für Information)

Anwendungs-spezifische Muster

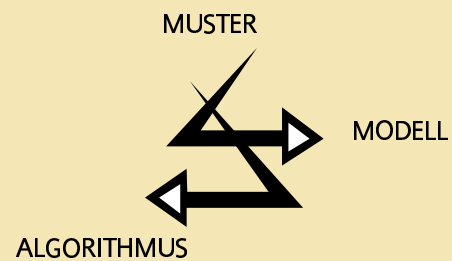
- Medizin
- Vermittlungssysteme & Telekommunikation (z.B. bei AT&T wurden Mehr als 100 Muster entwickelt)
- Lagerverwaltung
- Automatisierung

30.10.02

Ferienakademie 2002

19

BEZIEHUNGEN ZWISCHEN



30.10.02

Ferienakademie 2002

20

Schwierigkeiten (?)

- Gefahr der neuen Technik: „Blindheit“ für adäquatere Lösungen anstelle von Mustern -> Effizienzverlust
- hoher Lern- und Suchaufwand
- hoher Entwicklungsaufwand: kann nur von Personen mit sehr viel Erfahrung gemacht werden
- nicht automatisierbar (mit Ausnahme von Idiomen)

30.10.02

Ferienakademie 2002

21

Vorteile

- + Expertenwissen wird gesammelt und allgemein zugänglich gemacht
 - > Erhöhung des gesamten Niveaus möglich
- + Vereinfachte Kommunikation (Vorteile von Benennung)
- + Vielfalt der Grössenordnungen und Abstraktionsebenen
- + Vielfache Beziehungen zwischen den Mustern -> System
- + Mögliche Implementierung in jeder Programmiersprache
- + erfolgreiche Anwendungen in Geschäftswelt

30.10.02

Ferienakademie 2002

22

Informationen

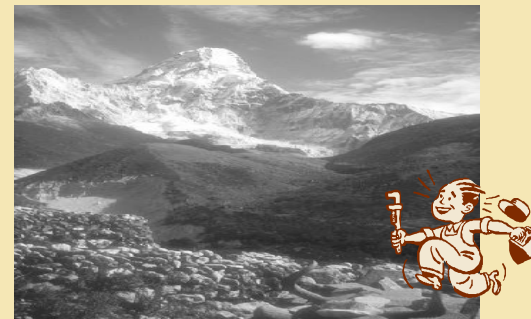
- Gang of Four – Erich Gamma, Richard Helm, Ralph Johnson und John Vlissides –
“DESIGN PATTERNS –Elements of Reusable Object-Oriented Software” 1994.
- Frank Buschman, Regine Meunier, Hans Rohnert, Peter Sommerlad, Michael Stal (Gang of Five oder GoV)
“ Pattern-orientierte Software-Architektur: A System of Patterns” (POSA) 1996.
- die Ausgaben der ersten und zweiten Konferenzen über Pattern Languages of Program Design (PLoP oder PLoPD)
<http://www.hillside.net>
<http://www.enteract.com/~bradapp/docs/patterns-intro.html>

30.10.02

Ferienakademie 2002

23

Ausblick



30.10.02

Ferienakademie 2002

24