

Entwurfsmuster in der objektorientierten Softwareentwicklung

Erika Mustermann

Matrikelnummer 54321

Exposé für die Lehrveranstaltung
Wissenschaftliches Arbeiten
im Studiengang
Betriebswirtschaft/Wirtschaftsinformatik

ASW-Gutachter
Prof. Dr. Dieter Hofbauer

Eingereicht am 4. Mai 2015

Inhaltsverzeichnis

1	Einleitung.....	1
2	Objektorientierte Softwareentwicklung	1
2.1	Objektorientierter Entwurf.....	1
2.2	Objektorientierte Implementierung.....	1
3	Unified Modelling Language.....	1
3.1	Klassen- und Objektdiagramme.....	1
3.2	Sequenzdiagramme	2
4	Entwurfsmuster	2
4.1	Architektur- versus Entwurfsmuster	2
4.2	Kategorisierung von Entwurfsmustern	2
5	Fazit.....	2
6	Literaturverzeichnis	3

1 Einleitung

Objektorientierte Softwareentwicklung ist aktueller Industriestandard, und Entwurfsmuster stellen Lösungen für immer wiederkehrende Problemstellungen beim Komponentenentwurf dar: „*The design patterns [...] are descriptions of communicating objects and classes that are customized to solve a general design problem in a particular context.*“¹ Das Buch von Gamma et al.² gilt als Meilenstein in der Historie des Softwareentwurfs. Seine Autoren³ beziehen sich bei der Wahl des Begriffs *Entwurfsmuster* explizit auf Alexander.⁴

2 Objektorientierte Softwareentwicklung

Die Rolle von Entwurfsmustern in Vorgehens- bzw. Prozessmodellen zur Softwareentwicklung wird ausführlich dargestellt.⁵ Zuletzt wird kurz auf die Verwendung von Entwurfsmustern im Kontext von Reengineering eingegangen.⁶

2.1 Objektorientierter Entwurf

Hier wird gezeigt, wie die allgemeinen Ziele des Softwareentwurfs durch konsequente Verwendung objektorientierter Grundkonzepte erreicht werden können.⁷

2.2 Objektorientierte Implementierung

Ein auf objektorientierten Konzepten basierender Entwurf kann bruchlos in jeder beliebigen objektorientierten Programmiersprache umgesetzt werden, etwa in der Sprache Java.⁸

3 Unified Modelling Language

Die Unified Modelling Language (UML) stellt unterschiedliche Modellierungsmittel in Form standardisierter Diagrammarten bereit.⁹ Bei der Beschreibung von Entwurfsmustern kommen vor allem Klassendiagramme¹⁰ und Sequenzdiagramme¹¹ zum Einsatz, seltener auch Objektdiagramme¹².

3.1 Klassen- und Objektdiagramme

Die für die Beschreibung von Entwurfsmustern wesentlichen Elemente eines Klassendiagramms sind einerseits die Strukturelemente (konkrete Klassen mit ihren Attributen und Operationen¹³ sowie abstrakte Klassen und Interfaces¹⁴), andererseits die Beziehungselemente

¹ Gamma et al., 1995, S. 3.

² Gamma et al., 1995, in der deutschen Übersetzung Gamma et al., 2011.

³ Die vier Autoren werden gelegentlich auch *Viererbande* (*Gang of Four*) genannt, vgl. Balzert, 2011, S. 37.

⁴ Alexander et al., 1977. Vgl. dazu auch die Erläuterungen bei Lea, 1994.

⁵ Vgl. Ludewig/Lichter, 2010, S. 154-177 zu Vorgehensmodellen und S. 181-234 zu Prozessmodellen.

⁶ Vgl. Chikofsky/Cross, 1990 und Ludewig/Lichter, 2010, S. 585-599.

⁷ Vgl. Ludewig/Lichter, 2010, S. 399-403 zur Entwurfsphase allgemein und S. 421-423 zur Objektorientierung.

⁸ Siehe Arnold et al., 2005 für eine fundierte Einführung.

⁹ Als Einführung und auch als Referenz eignen sich Oesterreich/Scheithauer, 2013 sowie Rupp et al., 2012. Zu den offiziellen Standards siehe OMG, o. J.

¹⁰ Vgl. Oesterreich/Scheithauer, 2013, S. 271-319, und Rupp et al., 2013, S. 101-163.

¹¹ Vgl. Oesterreich/Scheithauer, 2013, S. 360-364, und Rupp et al., 2013, S. 397-468.

¹² Vgl. Oesterreich/Scheithauer, 2013, S. 320, und Rupp et al., 2013, S. 177-187.

¹³ Vgl. Rupp et al., 2013, S. 108-120.

¹⁴ Vgl. Rupp et al., 2013, S. 122 f.

(Generalisierungen¹⁵ und Assoziationen¹⁶), die die Strukturelemente verbinden. Objektdiagramme dagegen verdeutlichen exemplarische Objektbeziehungen zu einem Zeitpunkt.¹⁷

3.2 Sequenzdiagramme

Im Gegensatz zur statischen Modellierung von Klassen- und Objektbeziehungen über Strukturiagramme bietet die UML auch zahlreiche Diagrammart zur Spezifikation der Systemdynamik. Neben den Aktivitätsdiagrammen¹⁸ und den Zustandsautomaten¹⁹ bieten sich im Kontext der Beschreibung von Entwurfsmustern vor allem Sequenzdiagramme²⁰ zur Verhaltensmodellierung an. Als Beispiele dienen hier die Muster *Beobachter*²¹ und *Befehl*²².

4 Entwurfsmuster

4.1 Architektur- versus Entwurfsmuster

Während Architekturmuster die Struktur eines Gesamtsystems und insbesondere die Kooperation zwischen seinen Subsystemen spezifizieren, behandeln Entwurfsmuster die Spezifikation der Struktur dieser Subsysteme; Entwurfsmuster werden daher gelegentlich auch als Mikroarchitekturen bezeichnet.²³ Als Beispiele für Architekturmuster werden das *Schichtenmuster*²⁴ sowie das *Model-View-Controller-Muster*²⁵ erläutert.

4.2 Kategorisierung von Entwurfsmustern

Die von Gamma et al. geprägten Kategorien (Erzeugungsmuster, Strukturmuster und Verhaltensmuster) werden erklärt und mit je einem Beispiel illustriert (*Abstrakte Fabrik*, *Kompositum* und *Beobachter*).²⁶ Zuletzt werden aktuelle Entwurfsmusterkataloge erwähnt.²⁷

5 Fazit

Nach einer Zusammenfassung der Arbeit wird als Ausblick noch kurz darauf eingegangen, dass viele der Entwurfsmuster aus Gamma et al. (1995) in aktuellen Klassenbibliotheken bereits umgesetzt sind. Dies wird am Beispiel der Programmiersprache Java anhand der Entwurfsmuster *Iterator*²⁸ und *Beobachter*²⁹ gezeigt, die sich in der Klasse *Iterator* bzw. den Klassen *Observer* und *Observable* wiederfinden.³⁰

¹⁵ Vgl. Rupp et al., 2013, S. 128-131.

¹⁶ Vgl. Rupp et al., 2013, S. 135-140.

¹⁷ Vgl. Oesterreich/Scheithauer, 2013, S. 320.

¹⁸ Vgl. Rupp et al., 2013, S. 259-324.

¹⁹ Vgl. Rupp et al., 2013, S. 325-396.

²⁰ Vgl. Rupp et al., 2013, S. 397-468.

²¹ Vgl. Gamma et al., 1995, S. 295.

²² Vgl. Gamma et al., 1995, S. 237.

²³ Vgl. Balzert, 2011, S. 37.

²⁴ Vgl. Balzert, 2011, S. 46-51, und Ludwig/Lichter, 2010, S. 430-432.

²⁵ Vgl. Balzert, 2011, S. 62-69, und Ludwig/Lichter, 2010, S. 433-435.

²⁶ Vgl. Gamma et al., 1995, S. 10.

²⁷ Als Beispiele sind Hillside Group, o. J. sowie Cunningham & Cunningham, o. J. geeignet.

²⁸ Vgl. Gamma et al., 1995, S. 257-271.

²⁹ Vgl. Gamma et al., 1995, S. 293-303.

³⁰ Vgl. Oracle, 2015 für die Schnittstellenspezifikation.

6 Literaturverzeichnis

- Alexander, Christopher/Ishikawa, Sara/Silverstein, Murray (mit Jacobson, Max/Fiksdahl-King, Ingrid/Angel, Shlomo) (1977): *A Pattern Language – Towns, Buildings, Construction*, New York: Oxford University Press, 1977
- Arnold, Ken/Gosling, James/Holmes, David (2005): *The Java Programming Language*, 4. Aufl., Boston u. a.: Addison-Wesley, 2005
- Balzert, Helmut (2011): *Lehrbuch der Softwaretechnik – Entwurf, Implementierung, Installation und Betrieb*, 3. Aufl., Heidelberg: Spektrum Akademischer Verlag, 2011
- Beck, Kent/Cunningham, Ward (1987): *Using Pattern Languages for Object-oriented Programs*, Technical Report No. CR-87-43, 1987, <http://c2.com/doc/oopsla87.html> (Zugriff 5.5.2014)
- Beck, Kent/Cunningham, Ward (1989): *A Laboratory for Object-oriented Thinking*, in: ACM SIGPLAN Notices, Vol. 24, Issue 10, 1989, S. 1–6
- Chikofsky, Elliot J./Cross, James H. (1990): *Reverse Engineering and Design Recovery – A Taxinomy*, in: IEEE Software, Vol. 7, Issue 1, 1990, S. 13–17
- Cunningham & Cunningham (Hrsg.) (o. J.): *Portland Pattern Repository's Wiki*, o. J., <http://c2.com/ppr/> (Zugriff 4.5.2014)
- Gamma, Erich/Helm, Richard/Johnson, Ralph/Vlissides, John (1995): *Design Patterns – Elements of Reusable Object-oriented Software*, Boston, MA: Addison-Wesley, 1995
- Gamma, Erich/Helm, Richard/Johnson, Ralph/Vlissides, John (2011): *Entwurfsmuster – Elemente wiederverwendbarer objektorientierter Software*, München: Addison-Wesley, 2011
- Hillside Group (Hrsg.) (o. J.): *Design Patterns Library*, o. J., <http://hillside.net/patterns/> (Zugriff 4.5.2014)
- Lea, Dough (1994): *Christopher Alexander – An Introduction for Object-oriented Designers*, in: ACM SIGSOFT Software Engineering Notes, Vol. 19, Issue 1, 1994, S. 39–46
- Ludewig, Jochen/Lichter, Horst (2010): *Software Engineering – Grundlagen, Menschen, Prozesse, Techniken*, 2. Aufl., Heidelberg: dpunkt Verlag, 2010
- Object Management Group (Hrsg.) (o. J.): *OMG Formally Released Versions of UML*, o. J., <http://www.omg.org/spec/UML/> (Zugriff 3.5.2014)
- Oesterreich, Bernd/Axel Scheithauer (unter Mitarbeit von Bremer, Stefan) (2013): *Analyse und Design mit der UML 2.5*, 11. Aufl., München: Oldenbourg Verlag, 2013
- Oracle (Hrsg.) (2015): *Java Platform API Specification*, Standard Edition 7, 2015., <http://docs.oracle.com/javase/7/docs/api/> (Zugriff 7.4.2015)
- Rupp, Chris/Queins, Stefan/die Sophisten (2012): *UML glasklar – Praxiswissen für die UML-Modellierung*, 4. Aufl., München: Carl Hanser Verlag, 2012