

Entwurfsmuster

WS 2006/2007



Themen, betreut von
Univ.-Prof. em. Dr. H.-J. Hoffmann

(Stand 18. Oktober 2006)

Die Literaturangaben in den einzelnen Themenblättern dienen als Anleitung zur jeweiligen Thematik. Die Bearbeiter müssen sich um einen darüber hinausgehenden Überblick bemühen.

Angaben noch unvollständig !

Alle Internet-Zugriffe August – Oktober 2006 !
Irrtum und Tippfehler vorbehalten !

Anleitung zur Vorbereitung Ihres Vortrags bzw. der Ausarbeitung:

- S.I.P. Jones et al.: *How to give a good research talk*,
ACM SIGPlan Notices 28 (1993) 11, 9 - 12
- M. Deininger et al.: *Studien-Arbeiten, ein Leitfaden ...*;
(u.a.) Teubner, 1992

**Beides kann in der Bibliothek des FB Informatik
eingesehen werden !**



Übersicht

- Einführung
 - #01 Anstoß aus Architektur, Muster alltäglich
 - #02 "GoF", Gamma et al.
 - #03 UML
- Entwurfsmuster für Programmiersprachen
 - #04 für Smalltalk
 - #05 für C++
 - #06 für Java
 - #07 für C#
 - #08 für Visual Basic .net *

- Entwurfsmuster bei Anwendungen
 - #09 für J2EE *
 - #10 Interaktionsmuster *
 - #11 bei Webentwurf *
 - #12 Echtzeitprogramme *
 - #13 *concurrent, parallel, distributed* *
 - #14 Entwicklungsprozesse *
- Sonstiges
 - #15 sog. *Anti-patterns* *

Mindestteilnehmerzahl 8.

Es werden **maximal 10 Themen** ausgegeben !

Die mit * markierten Themen stehen am Ende zur

Auswahl, Themen #01 bis #07 werden zuerst vergeben !

Anstoß aus Architektur, Muster alltäglich



- C. Alexander et al.: *A Pattern Language: Towns, Buildings, Construction* ; Oxford University Press, 1977
- J.O. Coplien, C. Alexander: *The Origins of Pattern Theory – the Future of the Theory, and the Generation of a living World* ; IEEE Software, 16 (1999) 5, 71 ff.
- J.O. Coplien, B. Appleton: *On the nature of „The Nature of Order“* ; Vortragsmitschrift, 1997 [35592]
- M.P. Cline: *The pros and cons of adopting and applying design patterns in the real world* ; Comm. ACM 39 (1996) 10, 47 – 49
- K. Klaus: *Das aktuelle Schlagwort: Entwurfsmuster* ; Informatik-Spektrum, 19 (1996) 5, 326 ff

Fortsetzung

Anstoß aus Architektur, Muster alltäglich



- D.C. Schmidt: *General Information on Patterns: 2004*
www.cs.wustl.edu/~schmidt/patterns-info.html
- M. Duell: *Non-Software Examples of Software Design Patterns*;
www2.ing.puc.cl/~jnavon/IIC2142/patexamples.htm
- R. Johnson: *Structure and Process in Patterns* ; Cincom Blog,
June 05, 2006,
[www.cincomsmalltalk.com/userblogs/ralph/blogview?
showComments=true&entry=3327001782](http://www.cincomsmalltalk.com/userblogs/ralph/blogview?showComments=true&entry=3327001782)
- (verschiedene, jährliche Sammelbände): *Pattern Languages of Program Design* ; Addison-Wesley, 1995 ff.

„Gang of four“ GoF, Gamma et al. Kategorien von Entwurfsmuster



- E. Gamma et al.^{§1}: *Design Patterns: Elements of reusable object-oriented software*; Addison-Wesley^{§2}, 1995 (und weitere, auch deutsch, kostenlos herunterzuladen von lci.cs.ubbcluj.ro/~raduking/Books/Design%20Patterns/)
- Erklärung der Abkürzung “GoF”, 2005, siehe hillside.net/patterns/DPBook/GOF.html
- Typisches, verbreitetes Beispiel (aus Smalltalk-Entwicklung, 1982) eines (zusammengesetzten) Entwurfsmusters, ursprünglich als Paradigma bezeichnet: *model-view-control* MVC, siehe Thema #04
- Beschreibung von Entwurfsmustern, “Entwurfsmustersprachen”, siehe Gamma et al., Abschnitt 1.3
- Wikipedia: en.wikipedia.org/wiki/Pattern_language

^{§1} Erich Gamma, Richard Holm, Ralph Johnson, John Vlissides†

^{§2} Der Verlag Addison-Wesley firmiert unterschiedlich.

„Gang of four“ GoF, Gamma et al. Kategorien von Entwurfsmuster



- *Creational*
- *Structural*
- *Behavioural*

- *Dynamic behavioural*
- *Foundational*
- *Process patterns*
- *Dynamic behavioural*
- *u.a*

There are 191 fundamental software patterns
(23 Gamma Patterns + 17 Buschmann Patterns + 72 Analysis
Patterns + 38 CORBA Design Patterns + 42 AntiPatterns)

entnommen aus:
www.antipatterns.com/briefing/sld001.htm

Universal Modelling language

UML



- Wikimedia: *Universal Modelling Language* ;
de.wikipedia.org/wiki/UML
- C. Craig: *UML 2 und Patterns angewendet* ; MITP-Verlag, 2005
- C. Larman: *Applying UML and Patterns* ; Prentice Hall, 2001
- D. Mapelsden et al.: *Design Pattern Modelling and Instantiation using DPML* ; Proc. 4th Intl. Conf. Tools Pacific: Objects for internet, mobile and embedded applications, 2002, 3 – 11 (über ACM Digital Library erhältlich)
- NN (Computer Technology Documentation Project): *UML Patterns* ; undated,
www.comptechdoc.org/independent/uml/begin/umlpatterns.html
- NN (Sparx-Systems): *UML Patterns* ; 2006,
www.sparxsystems.com/resources/developers/uml_patterns.html

Fortsetzung

Universal Modelling language

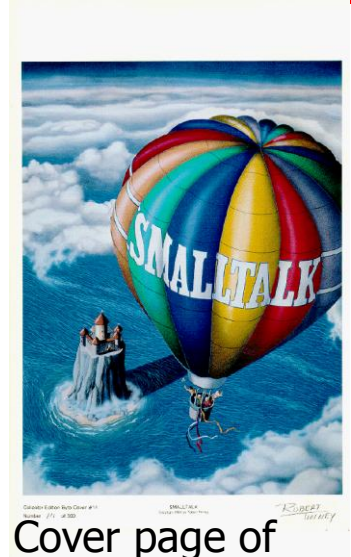
UML



- M. Grand: *Patterns in Java: A Catalog of Reusable Design Patterns illustrated with UML* ; Wiley, 2002 (vgl. Thema 06!)
- J.K.H. Mak et al.: *Precise Modeling of Design Patterns in UML* ; Proc. 26th Intl .Conf. Software Engineering, 2004, 252 - 261

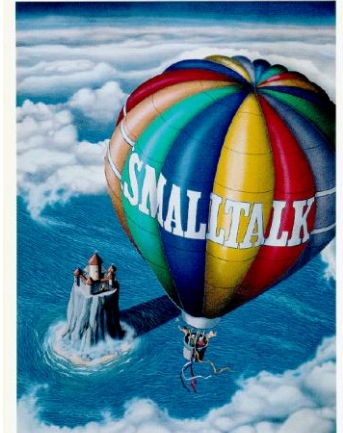


- A. Goldberg, D. Robson: *Smalltalk-80 – the Language*; Addison-Wesley, 1989
- Wikipedia: [http://de.wikipedia.org/wiki/Smalltalk_\(Programmiersprache\)](http://de.wikipedia.org/wiki/Smalltalk_(Programmiersprache))
- S.R. Alpert et al.: *The Design Patterns Smalltalk Companion*; Addison-Wesley, 1998
- K. Beck: *SmallTalk Best Practice Patterns*; Prentice Hall, 1996
- B. Foote: *A Smalltalk Patterns Safari* ; 1999/2006
www.laputan.org/talks/safari/safari.ppt
- S. Ducasse : *Free Online Books* ;
www.iam.unibe.ch/~ducasse/FreeBooks.html



Cover page of
August 1981 issue of
Byte magazine

Fortsetzung



Cover page of
August 1981 issue
of Byte magazine

- Typisches, verbreitetes Beispiel (aus Smalltalk-Entwicklung, 1982) eines Entwurfsmusters, ursprünglich als Paradigma bezeichnet: *model-view-control* MVC, siehe de.wikipedia.org/wiki/Model_View_Controller (siehe auch bei Gamma et al., Abschnitt 1.2 und T. Reenskaug: *The Model-View-Controller (MVC): Its Past and Present* ; 2003 heim.ifi.uio.no/~trygver/2003/javazone-jaoo/MVC_pattern.pdf)
- R. Johnson: *Model View Controller as an Aggregate Design Pattern* ; 2004, c2.com/cgi/wiki?ModelViewControllerAsAnAggregateDesignPattern

Entwurfsmuster für C++



- B. Stroustrup: *The C++ Programming Language* ; Addison-Wesley, 1986
- D. Vandevorde, N.M. Josuttis; *C++ Templates: The Complete Guide* ; Addison-Wesley, 2002
- N.M. Josuttis, D. Vandevorde; *C++ Templates: Metaprograms* ; Addison-Wesley, 2003
- NN (Addison Wesley): Auszug aus diesem Buch von Josuttis/Vandevorde, siehe www.awprofessional.com/articles/article.asp?p=30667&rl=1
- N.M. Josuttis, D. Vandevorde; *Templates and Inheritance Interacting in C++* ; Addison-Wesley, 2003
- NN (Addison Wesley): Auszug aus diesem Buch von Josuttis/Vandevorde, siehe www.awprofessional.com/articles/article.asp?p=31473&rl=1

Fortsetzung



- T. Muldner: *C++ Programming with Design Patterns Revealed* ; Addison-Wesley, 2002
- A. Alexandrescu: *Modern C++ Design: Generic Programming and Design Patterns Applied* ; Addison-Wesley, 2001
- D.C. Schmidt: *Object-oriented design case studies with patterns & C++* ; undatiert
www.cs.wustl.edu/~schmidt/PDF/pattern-examples4.pdf#search=%22c%2B%2B%20%22design%20pattern%22%22



- D. Flanagan: *Java in a Nutshell* ; O'Reilly, 1996 und spätere Auflagen (auch deutsch)
- M. Grand: *Patterns in Java* ; Wiley, vol. 1 (& 2), 2002 (1999)
- J.W. Cooper: *The design patterns Java companion* ; 1998, www.patterndepot.com/put/8/DesignJava.PDF
- J.W. Cooper: *Java™ Design Patterns: A Tutorial* ; Addison-Wesley, 2000
- P. Kuchana: *Software Architecture Design Patterns in Java* ; Auerbach, 2004
- S.J. Metsker, W.C. Wake: *Design Patterns in Java™* ; Addison-Wesley, 2006
- S.J. Metsker: *Design Patterns Java™ Workbook* ; Addison- Wesley, 2002

Fortsetzung





- S. Stelting, O. Maassen: *Applied Java™ Patterns* ; Prentice-Hall, 2001
- B. Brügge, A. H. Dutoit: *Objekt-orientierte Softwaretechnik mit UML, Entwurfsmustern und Java* ; Pearson Studium, 2004
- J. Hannemann, G. Kiczales: *Design pattern implementation in Java and AspectJ* ; ACM OPSLA, 2002, 161 - 173

Entwurfsmuster für C#



- P. Golde et al.: *Basic Concepts of the C# Programming Language*; Addison-Wesley, 2006
- NN (Addison-Wesley): *Auszüge aus dem Buch von Golde*; siehe www.awprofessional.com/articles/article.asp?p=474652&r1=1
- J.W. Cooper: *C# Design Patterns: A Tutorial*; Addison-Wesley, 2003
- NN (Addison-Wesley): *Auszüge aus dem Buch von Cooper*; siehe www.awprofessional.com/bookstore/product.asp?isbn=0201844532&r1=1
- NN: *Design Patterns in C# and .NET* (eine Zusammenstellung einiger Muster); undatiert, www.c-sharpcorner.com/Design.asp
- - Suche bei Google nach "*C# pattern*" bringt noch weitere Implementierungsanleitungen -

Entwurfsmuster für Visual Basic .Net



- M. Grand, B. Merrill: *Visual Basic Design Patterns* ; Wiley, 2005
- T. Fischer et al.: *Professional Design Patterns in VB .NET: Building Adaptable Applications* ; a! apress, 2006
- M. MacDonald: *Microsoft® Visual Basic® .NET Programmer's Cookbook* ; Microsoft Press, ISBN 0-7356-1931-X, 2003

www.microsoft.com/mspress/books/toc/6436.asp#TableOfContents

(das Wort „pattern“ habe ich dabei nicht gefunden, die reden von „recipes“ und „code snippets“ – na ja, Microsoft-typisch)

Entwurfsmuster für J2EE



- NN (SUN): *Core J2EE Patterns: Patterns index page* ; 2002, java.sun.com/blueprints/corej2eepatterns/Patterns/
- D. Alur et al.: *Core J2EE Patterns: Best Practices and Design Strategies, Second Edition* ; Prentice Hall, 2003
- D. Broemmer: *J2EE Best Practices: Java Design Patterns, Automation, and Performance* ; Wiley, 2002
- M. Grand: *Java Enterprise Design Patterns* ; Wiley, 2001
- B. Dudney: *J2EE AntiPatterns* ; Wiley 2003



T. Erickson: *The Interaction Design Patterns Page* ; 2004

www.visi.com/~snowfall/InteractionPatterns.html

T. Coram, J. Lee: *Experiences -- A Pattern Language for User Interface Design* ; 1996, siehe www.maplefish.com/todd/papers/experiences/Experiences.html

www.maplefish.com/todd/papers/experiences/Experiences.html

S. Fincher: *The Pattern Gallery* ; 2006

www.cs.kent.ac.uk/people/staff/saf/patterns/gallery.html

M.v. Welie: *GUI Design patterns* ; 2003

www.welie.com/patterns/gui/index.html

NN (Yahoo): *Design Pattern Library* ; 2006

developer.yahoo.com/ypatterns/

J. Borchers: *A Pattern Approach to Interaction Design* ; Wiley, 2001 (auch Dissertation TUD)

Fortsetzung



- J. Tidwell: *Designing Interfaces: Patterns for Effective Interaction Design* ; O'Reilly, 2005
- J. Tidwell: *Common Ground – A pattern language for human-computer interface design* ; MIT, 1999
www.mit.edu/~jtidwell/common_ground.html
- NN: *Patterns of Interaction: A Pattern Language for CSCW* ; undated,
www.comp.lancs.ac.uk/computing/research/cseg/projects/pointer/patterns.html
- NN: *Service interaction patterns* ; undated
sky.fit.qut.edu.au/~dumas/ServiceInteractionPatterns/



- E. Scheid et al.: *Website pattern* ; 2005, iawiki.net/WebsitePatterns
- M.v. Welie: *Web Design patterns* ; 2006 www.welie.com/patterns/
- R. Singer: *An Introduction to Using Patterns in Web Design* ; 2004, siehe www.37signals.com/papers/introtopatterns/
- M.Y. Ivory, R. Megraw: *Evolution of web site design patterns* ; ACM Trans. Information Systems, 23 (2005) 4, 463 - 497
- I. Graham: *A pattern language for Web usability*; Addison-Wesley ; 2003
- E. Scheid et al.: *Website patterns* ; 2005 www.iawiki.net/WebsitePatterns
- R. Orenstein: *An HTML 2.0 Pattern Language* ; 1995 www.anamorph.com/docs/patterns/index.html
(überholt, trotzdem noch interessant)

Entwurfsmuster für Echtzeitprogramme



- B.P. Douglass: *Real-Time Design Patterns: Robust Scalable Architecture for Real-Time Systems*; Addison Wesley, 2003
- NN (Eventhelix): *Embedded design patterns*; undated, www.eventhelix.com/RealtimeMantra/Patterns/
- J. Zalewski: *Real-time software design patterns*; 9th Polish Conf. on Real-Time Systems, 2002, 23 – 42, www-ece.engr.ucf.edu/~jza/publications/UstronFinal.pdf#search=%22ZALEWSKI%20pattern%22 (mit guten Literaturreferenzen)
- R. Sanz, J. Zalewski: *Pattern-based control systems engineering*; IEEE Control Systems, Vol. 23 (3), 2003, 43-60 www-ece.engr.ucf.edu/~jza/publications/SanzProof.pdf
- NN: *Real time patterns - use of object oriented design patterns* (in a telecommunication switch); undated, wiki.cs.uiuc.edu/cs427/Real+Time+Patterns



- B. Appleton: *Learning from Concurrent, Parallel, and Distributed Systems Design* ; 2004 www.cmcrossroads.com/component/option,com_magazine/func,show_article/id,155/Itemid,120/
(mit vielen relevanten Literaturstellen, u.a.:)
- D.C. Schmidt: *Patterns for Concurrent, Parallel, and Distributed Systems* ; OOPSLA 2004,
www.cs.wustl.edu/~schmidt/patterns-ace.html
- D. Kafura et al.: *Patterns for Concurrent, Parallel, and Distributed Systems* ; 1995 [delivery.acm.org/10.1145/270000/260266/p128kafura.pdf? key1=260266&key2=1636626511&coll=&dl=GUIDE&CFID=15151515&CFTOKEN=6184618#search=%22concurrent%20parallel%20distributed%20pattern%22](http://delivery.acm.org/10.1145/270000/260266/p128kafura.pdf?key1=260266&key2=1636626511&coll=&dl=GUIDE&CFID=15151515&CFTOKEN=6184618#search=%22concurrent%20parallel%20distributed%20pattern%22)
- T.G. Mattson et al.; *Patterns for Parallel Programming* ; Addison-Wesley, 2005

Anwendung von Mustern in Entwicklungsprozessen



- H. Züllighoven: *Object-oriented construction handbook* ; Elsevier, 2004
- NN (SPARX Systems): *Enterprise Architect – User Guide* ; 2006
www.sparxsystems.com/EAUserGuide/index.html?business_process_model_pattern.htm
- A. Rausch: *Prozessmusterbasierter Ansatz und der evolutionäre Entwurf* ; aus Dissertationsschrift TU München, 2001, Abschnitt 2.4, Seite 33 ff., agrausch.informatik.uni-kl.de/publikationen/repository/books/book001/componentware.pdf
- A. Förster et al.: *A pattern-driven development process for quality standard-conforming business process models* ; Proc. VL&HCC, Brighton, IEEE Press, 2006, 135 - 142

Fortsetzung

Anwendung von Mustern in Entwicklungsprozessen



- W.v.d.Aalst: *Workflow patterns* ; 2006
is.tm.tue.nl/research/patterns/
- W.M.P.v.d.Aalst et al.: *Workflow patterns* ; Distributed and Parallel Databases, vol. 14(3), 2003, 5 – 51,
is.tm.tue.nl/research/patterns/download/wfs-pat-2002.pdf
- F.J. Budinsky et al.: *Automatic code generation from design patterns* ; IBM Systems Journal, 35 (1996) 2, ??? [36058]
- J. Rajesh, D. Janakiram: *JIAD: a tool to infer design patterns in refactoring* ; Proc. 6th ACM SIGPLAN Intl. Conference Principles and Practice of Declarative Programming, 2004, 227 - 237

sog. *Antipattern*

- Wikipedia (deutsch): *Anti-Pattern* ; 2006
de.wikipedia.org/wiki/Anti-pattern
- Wikipedia (englisch): *Anti-pattern* ; 2006
en.wikipedia.org/wiki/Anti-pattern
- M.H. Dodani: *Patterns of Antipatterns* ; Journal of Object Technology, vol. 5, 2006, issue 6, 29 - 33
www.jot.fm/issues/issue_2006_07/column4
- W.J. Brown et al.: *Antipatterns – Refactoring Software, Architectures, and Projects in Crisis* ; Wiley, 1998 (book)
- H.W. McCormick: *Antipatterns – Refactoring Software, Architectures, and Projects in Crisis* ; Mitre Corp, 1998,
www.antipatterns.com/briefing/sld001.htm (slides)

Nachwort

If a software manual needs a book this thick,
then it is too complex for everyday use.



"When you design a language, you should not only consider what to put in, but also what to leave out." - Niklas Wirth

C++ has become an end to itself. Developers love it because they can tinker endlessly with the programming without ever coming close to solving a real-world problem.

Granted, Ada's language spec is even thicker. And while every surprise that you might encounter is properly documented, the sheer bulk ensures you won't know about it until you see for yourself.

Try Smalltalk. Just 20 pages to define the language. The rest of a Smalltalk book describes the IDE (the refactoring browser) and the libraries. Modern Smalltalks have everything that Java has, except the need for a "More Effective Smalltalk".

<http://books.slashdot.org/article.pl?sid=05/06/07/1531214&threshold=-1&tid=156&tid=6>

Und das gilt nicht nur für Programmiersprachen !

“Allgemeine Geschäftsbedingungen”



- Referat maximal 40 Minuten
- Datenprojektor („*beamer*“) ist vorhanden
- Anschließend Diskussion
 - inhaltlich
 - Vortragsstil
- Auf Anforderung, mindestens fünf Tage vorher, kann PC für Präsentation mit PowerPoint bereitgestellt werden
- Ausarbeitung maximal acht bis zehn Seiten DIN A4;
(in der Regel) nach einer Woche abzugeben (Papier);
Literaturangaben nicht vergessen!
- Nach Bestätigung (eventl. Korrekturwünsche) durch Betreuer
Abgabe der endgültigen Ausarbeitung innerhalb einer
Woche als **doc-/pdf-Datei**
(maximal 2 Mbyte groß, mit Namen “WS06-07-#xx-version-ii.<ext>”,
“xx” elem {01 ÷ 15}, “ii” elem {1 ...})

Ein Hinweis des Dekans:

“Der Fachbereich Informatik mißt der Einhaltung der Grundregeln der wissenschaftlichen Ethik großen Wert bei. Zu diesen gehört auch das strikte Verfolgen von Plagiarismus. Mit der Abgabe einer Lösung (Hausaufgabe, Programmierprojekt, Seminararbeit Diplomarbeit, etc.) bestätigen Sie, daß (Sie/Ihre Gruppe) der alleinige Autor/die alleinigen Autoren des gesamten Materials sind. Falls Ihnen die Verwendung von Fremdmaterial gestattet war, so müssen Sie dessen Quellen deutlich zitiert haben. Bei Unklarheiten zu diesem Thema finden Sie weiterführende Informationen unter www.informatik.tu-darmstadt.de/Plagiarism oder sprechen Sie Ihren Betreuer an.”



Web-Seite des Seminars:

<http://sun1.rbg.informatik.tu-darmstadt.de/PU/Seminar-Pattern>

(dort findet sich u.a. die Vorlage für das
Deckblatt Ihrer Ausarbeitung
und der **Terminplan** !)

**Nichts als Anstand, Sie beim
Lesen nicht stolpern zu lassen.**



HJH#??

2006/2007

<Thema>



- <offen>