8. Fachgruppentreffen der GI-Fachgruppe 2.1.9 Objektorientierte Software-Entwicklung

NetObjectDays 2001, Messe Erfurt, 11. September 2001 16:30 Uhr

Anliegen

Die Fachgruppe Objektorientierte Software-Entwicklung ist eine der Fachgliederungen der Gesellschaft für Informatik. Sie verfolgt das Ziel, Anwendern aus der Industrie und Forschenden und Lehrenden an Hochschulen ein Forum zu bieten, um den Einsatz objektorientierter Technologien zu unterstützen und die Aktivitäten einzelner zu bündeln. Dazu gehören z.B. die gemeinsame Arbeit an neuen Konzepte und Standards und das Verfolgen internationaler Entwicklungen auf dem Gebiet der Objektorientierung. Für spezielle Themen existieren Arbeitskreise, die teilweise enge Formen der Zusammenarbeit realisieren. Für aktuelle Informationen unterhält die Fachgruppe eine Website unter http://www.upb.de/cs/GI-FG219/ .

Das Fachgruppentreffen findet jährlich am Rande einer Fachkonferenz statt. Es dient dazu, den Kontakt zwischen allen Mitgliedern aufrecht zu erhalten, die Ergebnisse der Arbeitskreise vorzustellen, einen Überblick über aktuell von Fachgruppenmitgliedern bearbeitete Themen zu geben sowie Entscheidungen über die künftige Tätigkeit der Fachgruppe zu treffen. Auch Interessenten, die nicht Mitglied der Fachgruppe sind, sind als Gäste herzlich eingeladen.

Programm und Zeitplan

16:30 Begrüßung

Vorträge mit anschließender Diskussion

16:40 P. Fettke, P. Loos: Zur Klassifikation von Patterns

17:20 S. Busse u.a.: Domain Engineering informationslogistischer Anwendungen

18:00 Bericht über die Arbeit der Fachgruppe und der Arbeitskreise

18:40 Diskussion über zukünftige Aktivitäten

Ende vorr. 19:20 Uhr

Ansprechpartner

Gregor Engels, Universität Paderborn (Sprecher der FG-Leitung)
Matthias Riebisch, TU Ilmenau (Organisation dieses Treffens)

Zur Klassifikation von Patterns

Peter Fettke, Peter Loos

Technische Universität Chemnitz, Fakultät für Wirtschaftswissenschaften, Information Systems & Management; D-09107 Chemnitz, Germany, Tel.: +49/371/531-4375, Fax: -4376, E-Mail: {peter.fettke|loos}@isym.tu-chemnitz.de; WWW: http://www.isym.tu-chemnitz.de/

„Patterns are proven solutions to recurring design problems.“[Copl00, S. 1604] In der Literatur sind inzwischen zahlreiche Muster dokumentiert. Exemplarisch sei hier auf die Übersicht von [Risi00] verwiesen, in der über 1000 Muster nachgewiesen sind. Es ist offensichtlich, daß eine rein sequentielle Suche in dieser Mustermenge nicht effizient ist. Vielmehr wird ein Ordnungssystem benötigt, um für spezifische Problemstellungen geeignete Muster wiederaufzufinden.

In dem Beitrag werden zunächst Anforderungen an ein Ordnungssystem für Muster entwickelt. Ein Ansatz sollte formal-wissenschaftlichen Kriterien wie Vollständigkeit, Präzision und Konsistenz genügen. Ferner sollte zum einen die Erweiterbarkeit und zum anderen die Benutzbarkeit des Ordnungssystems gewährleistet sein. Schließlich sollte die Erstellung und Anwendung eines Ordnungssystem dem Prinzip der Wirtschaftlichkeit unterliegen. Zwischen diesen Anforderungen bestehen Zielkonflikte.[FeLo00]

Im zweiten Teil des Beitrages werden verschiedene Ordnungssysteme für Muster vorgestellt. Die präsentierten Ansätze umfassen einerseits Arbeiten, die sich direkt mit der Klassifikation von Mustern beschäftigen.[bspw. Tich97; Risi00] Anderseits werden auch Ordnungssysteme beschrieben, die von den Autoren bekannter Musterbücher verwendet werden.[bspw. GHJV95; CoSc95; Busc96; SSRB00]

Der dritte Teil des Beitrages bewertet die beschriebenen Ordnungssysteme im Hinblick auf die zuvor erarbeiteten Anforderungen. Es zeigt sich, daß die bisher verwendeten Ordnungssysteme wenig befriedigen. Beispielsweise werden in [Risi00] betriebswirtschaftliche Kategorien wie Banking oder Accounting technischen Merkmalen wie Analysis, GUI Development ohne nähere Erläuterung gleichgeordnet. Abgeschlossen wird der Beitrag mit Überlegungen, wie ein leistungsfähiges Ordnungssystem für Muster ausgestaltet sein könnte. Beispielsweise könnten Muster hinsichtlich verschiedener Aspekte wie fachliche Anwendungsdomäne im Allgemeinen, betriebswirtschaftliche Anwendungsdomäne im Besonderen, technische Anwendungsdomäne, Phase in der Softwareentwicklung und Implementierungssprache klassifiziert werden. Eine derartige Aspekt-basierte Klassifikation ist ein vielversprechender Ansatz zur Systematisierung von Mustern.

Literatur

[Busc96] Buschmann, F.; Meunier, R.; Rohnert, H.; Sommerlad, P.; Stal, M.: Pattern-Oriented Software Architecture - A System of Patterns. Chichester et al. 1996.

[Copl00] Coplien, J. O.: Software Design Patterns. In: A. Ralston; E. D. Reilly; D. Hemmendinger (Hrsg.): Encyclopedia of Computer Science. London, New York, NY, 2000, S. 1604-1606.

[CoSc95] Coplien, J. O.; Schmidt, D. C.: Pattern Languages of Program Design. Reading, MA, et al. 1995.

[FeLo00] Fettke, P.; Loos, P.: Komponentendokumentationen - Eine systematische Bewertung von Ordnungssystemen aus formaler Sicht. In: K. Turowski (Hrsg.): Modellierung und Spezifikation von Fachkomponenten: Workshop im Rahmen der MobIS 2000 Modellierung betrieblicher Informationssysteme, Siegen, Deutschland, 12. Oktober 2000, Tagungsband. Siegen 2000, S. 51-70.

[GHJV95] Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J.: Design Patterns - Elements of Reusable Object-Oriented Software. Reading, MA, et al. 1995.

[Risi00] Rising, L.: The Pattern Almanac 2000. Boston et al. 2000.

[SSRB00] Schmidt, D.; Stal, M.; Rohnert, H.; Buschmann, F.: Pattern-Oriented Software Architecture - Patterns for Concurrent and Networked Objects. Chichester et al. 2000.

[Tich97] Tichy, W. F.: A Catalogue of General-Purpose Software Deisgn Patterns. In: IEEE Computer Society (Hrsg.): Tools-23: Technology of Object-Oriented Languages and Systems, Santa Barbara, California, USA, 28 July - 1 August, 1997. Santa Barbara, California, USA 1997

Domain Engineering informationslogistischer Anwendungen

Susanne Busse, Gernot Goldbeck, Nieves Lozano,
Johannes Nold, Wolfgang Sigel
TU Berlin, CIS

Die Fachgruppe 'Computergestützte Informationssysteme' (CIS) der TU Berlin erarbeitet seit einigen Jahren in Kooperation mit dem Kompetenzzentrum Informationslogistik des Fraunhofer-Instituts für Software- und Systemtechnik ISST Konzepte zur Entwicklung informationslogistischer Anwendungen ([1], [2]).

Die Informationslogistik (ILOG) hat das Ziel, die individuelle Informationsversorgung zu verbessern, indem die spezifischen Bedürfnisse bzgl. der ILOG-Dimensionen Inhalt, Zeit, Ort und Präsentation berücksichtigt werden. Beispiel-Anwendungen, die am Fraunhofer ISST entwickelt wurden, sind etwa das Unwetterfrühwarnsystem '@ptus weather', der Weckdienst 'w@ke up', der die aktuelle Stausituation berücksichtigt, oder der Abonnement-Dienst '@ptus news'.

Basis für eine schnelle Entwicklung solcher Anwendungen bildet der Ansatz des Domain Engineerings ([3]). ILOG-Anwendungen werden als eine Systemfamilie aufgefasst, denen eine gemeinsame Architektur sowie gemeinsame Basiskomponenten zugrundeliegen. Die Varianten ergeben sich durch die spezifische Konfiguration der Komponenten, die bestimmt, in welcher Art und Weise die ILOG-Dimensionen gewichtet und kombiniert werden. Das anwendungsspezifische Zusammenspiel der Komponenten bezeichnen wir als ILOG-Kernprozess. Er umfasst die Bedarfsangabe eines Nutzers, die Überwachung relevanter Ereignisse und die Informationsermittlung, -aufbereitung und -zustellung.

Die Entwicklung wird konkret durch das Werkzeug GENIA (Generative Entwicklung informationslogistischer Anwendungen) unterstützt. Es beinhaltet ein Feature Modell der Varianten von ILOG-Anwendungen und ihre Auswirkungen auf den ILOG-Kernprozess. Ausgehend von einer Spezifikation der gewünschten Eigenschaften einer ILOG-Anwendung kann so ein erster ILOG-Kernprozess, d.h. eine geeignete Konfiguration der Komponenten abgeleitet werden. Der Prozess wird als UML Activity Diagramm dargestellt und kann manuell an konkrete Gegebenheiten angepasst werden. Aus dem endgültigen Modell werden automatisch die den Prozess realisierenden Software-Komponenten generiert.

[1] Informationslogistik bei der Fachgruppe CIS,
http://cis.cs.tu-berlin.de/Forschung/Projekte/ilog.html

[2] Kompetenzzentrum Informationslogistik des Fraunhofer ISST,
http://www.informationslogistik.org/

[3] K. Czarnecki, U.W. Eisenecker, Generative Programming - Methods, Tools, and Applications, Addison-Wesley, 2000.