Protokoll des 4. Fachgruppentreffens der GI-Fachgruppe 2.1.9 "Objektorientierte Software-Entwicklung"

am 6. Februar 1997 in München, Sheraton-Hotel

Protokoll: Gregor Engels

Beginn: 9:45 Uhr **Ende:** ca. 18 Uhr

Das 4. Fachgruppentreffen der GI-Fachgruppe 2.1.9 fand im Rahmen der OOP'97/Java Expo im Sheraton-Hotel in München statt. Frances Paulisch, fachliche Leiterin der OOP'97/Java Expo und auch Mitglied des Leitungsgremiums der FG 2.1.9, hatte es möglich gemacht hat, daß der FG ganztägig ein Raum zur Verfügung stand. Hierzu sei ihr auch an dieser Stelle gedankt.

Die im Oktober 1995 gegründete FG umfaßt mittlerweile mehr als 250 Mitglieder.[1] 55 von ihnen waren zum FG-Treffen erschienen, wobei erfreulicherweise die Hälfte aus der Industrie und die andere Hälfte aus dem universitären Umfeld kam.

Das Programm bestand aus 4 Teilen, über die im folgenden kurz berichtet wird.

TOP 1: Berichte aus den Arbeitskreisen

Hauptziel des FG-Treffens war, den zu diesem Zeitpunkt 4 aktiven Arbeitskreisen Gelegenheit zu geben, ihren aktuellen Diskussionsstand der gesamten Fachgruppe zu präsentieren. Hierzu hatte jeder Arbeitskreis 1 - 2 Sprecher benannt, die dann in einem etwa halbstündigen Vortrag mit anschließender Diskussion über die aktuelle Arbeit in den Arbeitskreisen berichteten.

Im einzelnen wurden die folgenden Vorträge gehalten. Zusammenfassungen der Vorträge findet man am Ende dieses Protokolls.

Fachvorträge AK Frameworks

Thomas Stalzer: Das Anwendungsframework VASER

Rolf Knoll: Gebos Rahmenwerke für die Softwareentwicklung im Bankenbereich

Fachvortrag AKVO (Verteilte Objekte)

Fridtjof Toenniessen, Andreas Wickner: Eine Mustersprache für CORBA in großen betrieblichen Informationssystemen

Fachvorträge AK GROOM (Grundlagen objektorientierter Modellierung)

Roger Süttenbach, Universität Koblenz-Landau: Formalisierung der Syntax von objekt-orientierten Methoden

Cornel Klein, Bernhard Rumpe, Technische Universität München: Formalisierung der Semantik objekt-orientierter Methoden

Fachvorträge AK Qualitätsverbesserung

Jürgen Znotka, Fachhochschule Lübeck: Produkt und Prozeßmodelle der objektorientierten Softwareentwicklung als Basis einer effektiven Softwarequalitätssicherung

Jinhua Li, Universität Stuttgart: Wartungsunterstützung für Smalltalk-Anwendungen durch statische Programmanalyse

Die Vorträge zeigten in beeindruckender Weise, wie breit das Gebiet der objektorientierten Softwareentwicklung mittlerweile geworden ist. Alle aktuellen Schlagworte wie z.B. Entwurfsmuster, Framework, CORBA, Proxy, UML oder Smalltalk kamen zur Sprache und es wurde anhand einer Reihe konkreter Projekte bzw. Produkte gezeigt, wie diese Begriffe in der Praxis zum Einsatz kommen. Da nahezu alle Vorträge sehr technisch und konkret waren, war die anschließende Diskussion sehr konkret. Der allgemeine Eindruck war, daß dadurch die aktuellen Probleme bei einer objektorientierten Vorgehensweise sehr deutlich wurden. Alles in allem waren dadurch die Vorträge zusammen mit der Diskussion für alle Beteiligten sehr interessant und gewinnbringend.

TOP 2: Vorstellung eines neuen Arbeitskreises

Herr Krzystof Czarnecki vom Daimler-Benz Forschungszentrum in Ulm präsentierte erste Ideen für einen neuen Arbeitskreis zum Thema "Generative und komponentenbasierte Softwareentwicklung". Er erläuterte die Grundkonzepte einer derartigen Softwareentwicklung und macht den Zusammenhang deutlich zu existierenden Konzepten wie z.B. parametrisierte Typen, Adaptive Programming, Transformationssysteme oder Intentional Programming. Weitere Initiatoren dieses Arbeitskreises sind Ulrich Eisenecker (FH Heidelberg) und Manfred Broy (TU München).

Die FG hat diese Initiative erfreut begrüßt. In-

[1] Nähere Informationen zur FG erhält man über die Homepage der FG (http://www.wi.leidenuniv.nl/~zamper/gi-fg-2.1.9.html).

teressenten an diesem AK können näheres über die Homepage des AK erfahren (http://www.prakinf.tu-ilmenau.de/~czarn/generate)
oder sich direkt an Herrn Czarnecki wenden (czarnecki@dbag.ulm.DaimlerBenz.com).

TOP 3: Weiteres Vorgehen

Eine Fachgruppentreffen am Rande der OOP-Konferenz wurde von allen Beteiligten sehr begrüßt. Dies gab den Teilnehmern die zusätzliche Möglichkeit, an der OOP teilzunehmen oder auch nur (kostenfrei) einige der eingeladenen Vorträge zu hören und die interessante und umfangreiche Ausstellung zu besuchen. Es war einhellige Meinung zu versuchen, im nächsten Jahr eine derartige Veranstaltung zu wiederholen. Andererseits war man auch der Meinung, daß ein FG-Treffen einmal pro Jahr ausreichend ist. Die inhaltliche Arbeit kann und wird in den Arbeitskreisen getan, die sich häufiger pro Jahr treffen.
So trafen sich anschließend an dieses FG-Treffen auch die beiden Arbeitskreise Frameworks und GROOM zu einem ein- bis zweitägigen Treffen an der TU München. Die Protokolle zu diesen beiden Treffen findet man bereits in der letzten Ausgabe der Softwaretechnik-Trends.

TOP 4: Social Event

Der Tag wurde abgeschlossen mit einem gemütlichen Abendessen im "Donisl" direkt am Marienplatz in der Innenstadt München. Alles in allem ein sehr gelungener Tag in München!

Zusammenfassungen der Vorträge

Das Anwendungsframework VASER

Thomas Stalzer, OBJECT *dynamics* Software GmbH, Stuttgart

VASER ist ein auf dem Visual Age/Smalltalk-System aufbauendes, kommerziell verfügbares Framework, das im Unterschied zu rein bibliotheksgestützten objektorientierten Frameworks auch stark auf einer eigenen Werkzeugumgebung (Editoren und Generatoren) basiert. VASER definiert eine 3-Schichtenarchitektur, bestehend aus Anwendungsschicht (Präsentationselemente und einfache Dialogelemente), Unternehmensschicht (Unternehmensmodell, Geschäftsobjekte) und Systemschicht (Datenbankankopplung). Zur Entkopplung der Schichten benutzt VASER das Part/Wrapper-Konzept von Visual Age. Aus dem Unternehmensmodell läßt sich eine Standard-Benutzerschnittstelle zur Anzeige bzw. Bearbeitung von Objektattributen und ein Datenbankschema generieren. Darüberhinaus definiert VASER Transaktionen zur Erhaltung der Konsistenz zwischen den drei Schichten. Zur Realisierung persistenter Objekte dient ein eigenes Objekt-Repository.

Gebos Rahmenwerke für die Softwareentwicklung im Bankenbereich

Rolf Knoll, RWG Stuttgart

GEBOS (Genossenschaftliches Büro- und Organisationssystem), zur Zeit aus 1800 C++ Klassen bestehend, stellt Frameworks für die Entwicklung von Beraterarbeitsplätzen im Bankenbereich zur Verfügung, die bei Genossenschaftsbanken bereits erfolgreich im Einsatz sind. GEBOS basiert auf der Entwurfsmetapher Werkzeug-Aspekt-Material. Bei den Werkzeugen sind Funktionalität und Interaktion getrennt. Es existieren zwei Klassenhierarchien, eine für fachliche Werte (z.B. Kontonummer, Zinssatz) und eine für die entsprechenden Interaktionstypen (Kleinsteditoren für diese Werte). Ein Entwurfsmuster zur flexiblen, späten Erzeugung verringert die Kopplung zwischen den entsprechenden Klassen aus beiden Hierarchien. Jeder variable Aspekt, Konzept genannt, wird auf der konzeptionellen Schicht durch eine abstrakte Klasse definiert, die Basisschicht enthält als Unterklasse mindestens eine Standardimplementierung, und die Projektschicht enthält projektspezifische Subklassen.

Eine Mustersprache für CORBA in großen betrieblichen Informationssystemen

Fridtjof Toenniessen, Andreas Wickner, sd&m, München

Im Rahmen des Projekts DaRT wurde von sd&m und der Deutschen Bahn AG ein modernes, CORBA-basiertes Verwaltungs- und Auskunftssystem für die Fahrzeugdaten der Bahn-Geschäftsbereiche Fernverkehr, Nahverkehr und Traktion entwickelt. Als Problem stellte sich dabei die effiziente Übermittlung des Datenanteils komplexer Objekte in Netzwerken mit geringer Bandbreite heraus. Als Lösung wurde eine neue Mustersprache "Smart Proxy Decomposition" entwickelt, die basierend auf den Grundmustern "Proxy" und "Strategy" sowie einer geeigneten Strukturierung der IDL-Spezifikation, eine Übertragung und Veränderung von Instanzdaten erlaubt, ohne dafür nicht-standardisierte CORBA-Erweiterungen zu erfordern. Die konsequente Anwendung von Generatoren reduziert den Entwicklungsaufwand hierbei erheblich.

Formalisierung der Syntax von objektorientierten Methoden

Roger Süttenbach, Universität Koblenz- Landau

In diesem Vortrag wird der in Koblenz entwickelte EER/GRAL-Ansatz (Extended Entity-Relationship/GRAph specificiation Language) zur Formalisierung von objektorientierten Methoden verwendet. Durch die Beschreibung mittels ER-Diagrammen und

GRAL-Prädikaten ergibt sich eine präzise und integrative Darstellung. Dabei beschränkt sich die Formalisierung zur Zeit auf die abstrakte Syntax der in einer Methode verwendeten Modellierungssprachen inklusive zugehöriger Konsistenzbedingungen.
Formalisierungen fuer die Booch-Methode und OMT-Methode wurden bereits erarbeitet. Dabei wurde besonderes Gewicht auf die Vollständigkeit der Beschreibung hinsichtlich der verwendeten Konzepte und Konsistenzbedingungen gelegt. Die Mittel, Prinzipien und die Vorgehensweise zur Erstellung einer solchen Formalisierung werden an einem anschaulichen Beispiel aus der Booch Methode verdeutlicht.

Formalisierung der Semantik objekt-orientierter Methoden

Cornel Klein, Bernhard Rumpe, Technische Universität München

Objektorientierte Analyse- und Designmethoden erlauben die Modellierung eines Systems auf verschiedenen Abstraktionsebenen und unter verschiedenen Sichten. Während für einzelne Beschreibungstechniken (wie z.B. ER-Modelle, Automaten) teilweise relativ präzise, formal definierte, Semantiken existieren, fehlt bislang eine integrierte Semantik für die Beschreibungstechniken einer Methode. Diese ist notwendig für semantikerhaltende Transformationen zwischen Dokumenten, für umfangreiche Konsistenzüberprüfungen, sowie für die Erstellung fortgeschrittener Werkzeugunterstützung für die Generierung und Analyse.
Wir stellen einen Ansatz zur formalen Definition einer solchen integrierten Semantik vor. Dieser baut auf einem mathematischen Systemmodell auf, welches zentrale Konzepte informationsverarbeitender Systeme wie *Komponente, Nachricht, Komposition* etc. formal faßt. Zu diesem mathematischen Systemmodell können alle Beschreibungstechniken einer Software Engineering-Methode in Beziehung gesetzt werden. Wir berichten über Erfahrungen, die wir bei einer derartigen Formalisierung eines Ausschnitts der Unified Modelling Language (UML) gewonnen werden.

Produkt- und Prozeßmodelle der objektorientierten Softwareentwicklung als Basis einer effektiven Softwarequalitätssicherung (SQS)

Jürgen Znotka, Fachhochschule Lübeck

In einer effektiven SQS werden Prozesse und Produkte einer analytischen Beurteilung unterzogen. Während Prozesse in den letzten Jahren vielfach untersucht und verschiedene Ansätze zu ihrer Verbesserung entwickelt wurden (z.B. ISO 9000, CMM), ist dies bei Produkten nicht in dieser Ausführlichkeit geschehen. Basili merkte dazu an, daß Produktqualität offenbar schwerer zu erreichen sei als Prozeßqualität. Ein Problem ist sicher der komplexe Zusammenhang zwischen Prozessen und ihren Produkten, der eine Betrachtung der Produkte hinsichtlich ihrer Qualität erschwert.

Durch die Einführung atomarer Prozesse in der Modellierung wird eine Abstraktionsebene geschaffen, die es ermöglicht, bei einer definierten Entwicklungsmethodik (z.B. OMT, UML) beliebige konkrete Prozessmodelle in Rahmen der Entwicklung einzusetzen. Das aus der Methodik abgeleitete Produktmodell kann dabei unabhängig vom konkreten Prozessmodell im Rahmen der Qualitätssicherung verwendet werden.

Wartungsunterstützung für Smalltalk- Anwendungen durch statische Programmanalyse

Jinhua Li, Universität Stuttgart

Die durch statische Programmanalyse ermittelten Informationen können auch für die Softwarewartung verwendet werden. Allerdings ist die herkömmliche Programmanalyse nicht direkt auf Smalltalk anwendbar, da die Datenflußanalyse und die Kontrollflußanalyse voneinander abhängig sind. Außerdem erschwert die dynamische Typisierung in Smalltalk eine derartige statische Codeanalyse. In dem Vortrag wurde eine Methode zur statischen Programmanalyse vorgestellt. Der Kern dieser Methode ist eine constraintbasierte Typinferenz. Die Bestandteile der Typinferenz, ein Typsystem, Typconstraints und ein Inferenzalgorithmus wurden erklärt. Anhand eines Prototyps wurde gezeigt, wie die gewonnenen Informationen die Wartungsaufgaben unterstützen können.