Protokoll des 7. Fachgruppentreffens der GI-Fachgruppe 2.1.9 Objektorientierte Software-Entwicklung

am 26.1.2000 in München, ICM, OOP 2000

Protokoll: Silke Seehusen

http://www.upb.de/cs/ag-engels/GI/gi-fg219.htm

Tagesordnung

TOP 1: 9:00 - 9:05, **Begrüßung**
Gregor Engels (FG-Leitung)

TOP 2: 9:05 - 11:00 Fachvorträge I
Matthias Riebisch, Detlef Streitferdt, Kai Böllert (TU Ilmenau): Methoden und Werkzeuge zur Entwicklung von Systemfamilien
Holger Giese, Joerg Graf, Guido Wirtz (Universität Münster): Modellierung verteilter Systeme nach dem OCoN/UML-Ansatz
Cay Kaben, Silke Seehusen, Jens Seehusen (FH Lübeck, innovas Hamburg): Agentensystem zur Unterstützung von Geschäftsvorfällen im Versicherungswesen
Mario Jeckle (DaimlerChrysler Forschungszentrum Ulm): XML Schemata – Notwendigkeit, Ansätze, Standards

TOP 3: 11:15 - 12:00, **OOP 2000 Keynote**
Blake Stone (Inprise Corp.): Open Source meets Commercial Software

TOP 4: 14:30 - 15:00, **Fachvorträge II**
Arne Koschel (IONA Professional Services): Die Basis von IONA's iPortal Suite - iPortal Applicaiton Server and Orbix 2000

TOP 5: 15:00 - 16:00, **Arbeitskreise der FG 2.1.9**
Vorstellung aktueller Aktivitäten, Neugründung von Arbeitskreisen

TOP 6: 16:15 - 17:00, **Neuwahl des Leitungsgremiums, Abschlußdiskussion**

TOP 7: 17:15 - 18:00, **OOP 2000 Keynote**
Gero Scholz (Dresdner Bank): Der Beitrag von Java und XML zur Weiterentwicklung der Anwendungslandschaft

TOP 1: Begrüßung

Das 7. Fachgruppentreffen wurde von Gregor Engels eröffnet und von Silke Seehusen geleitet. Auch das diesjährige Treffen der Fachgruppe 2.1.9 fand wieder im Rahmen der OOP in München statt. Es waren ca. 40 Personen anwesend, die wie auch schon in den letzten Jahren zur Hälfte aus der Industrie und zur Hälfte von der Hochschule kamen. Aufgrund des Einsatzes von Frau Paulisch (Mitglied der Fachgruppenleitung) und der freundlichen Unterstützung durch den Veranstalter SIGS Conferences stand auch in diesem Jahr wieder kostenlos ein Raum im neuen Konferenzzentrum ICM zur Verfügung.

TOP 2 und 4: Fachvorträge

Es wurden aktuelle Arbeiten von Mitgliedern der Fachgruppe präsentiert. Zu den einzelnen Vorträgen sei auf die Kurzfassungen am Ende dieses Protokolls verwiesen. Es fand eine sehr rege Diskussion zu den einzelnen Themen statt.

TOP 3 und 6: OOP 2000 Keynotes

Im Rahmen der OOP fanden die zwei Vorträge von Blake Stone und Gero Scholz statt. Desweiteren war der Besuch der Ausstellung der OOP 2000 vorgesehen.

Top 5: Arbeitskreise der FG 2.1.9

Es wurden die bisherigen und geplanten Aktivitäten der Fachgruppe von S.Seehusen berichtet. Es folgten Berichte aus den Arbeitskreisen

- AK GROOM, A. Schürr
- AK Verteilte Objekte, C. Zeidler
- AK Frameworks, M. Riebisch, R. Knoll
- AK Generative Software-Entwicklung, E. Pulvermüller

Die Arbeitskreise sind sehr aktiv und beteiligen sich an Workshops und Konferenzen. Es finden mehrere Treffen pro Jahr statt (siehe auch http://www.upb.de/cs/ag-engels/GI/gi-fg219.htm).

Es wurden die Themen eines neu zu gründenden Arbeitskreises zum Thema Agententechnologie von S.Seehusen vorgestellt. Dieser Arbeitskreis wurde mehrheitlich begrüßt. Zu einem ersten Treffen wird dieses Jahr über die oose-email-Liste eingeladen.

Der Arbeitskreis Qualitätssicherung und Testen wird aufgelöst. Aus dem Arbeitskreis ist die neue

Fachgruppe 2.1.10 Software-Messung und Bewertung hervorgegangen. Zum Thema Testen von OO Programmen gibt es einen Arbeitskreis in der Fachgruppe 2.1.7. Ein Verweis auf diesen Arbeitskreis wird auf der WWW-Seite der FG 2.1.9 aufgenommen.

TOP 6: Neuwahl des Leitungsgremiums und Abschlußdiskussion

Die Neuwahl der Fachgruppenleitung (FGL) wurde von A. Zamperoni als Wahlleiter durchgeführt. Dazu gibt es ein eigenes Wahlprotokoll des Wahlleiters. Die alte FGL wurde beauftragt, eine Satzungsänderung der FG in der Hinsicht zu beantragen, dass die FGL ab jetzt aus 5 Mitgliedern besteht. Als Mitglieder der FGL wurden gewählt:

1. Gregor Engels,
2. Silke Seehusen,
3. Frances Paulisch,
4. Christian Zeidler und
5. Matthias Riebisch.

In der Abschlußdiskussion wurde ein erweiterter Aufgabenbereich der Fachgruppenleitung diskutiert. Die Aufgaben der FGL sollen umfassen

- Repräsentation innerhalb der GI (organisatorisches, Vertretung der FG in den GI-Organen),
- Repräsentation der Fachgruppe und des Themas OOSE in der fachlichen Öffentlichkeit, u.a. Beiträge in den Softwaretechnik-Trends veröffentlichen (z.B. ausgewählte Vorträge der FG- und AK-Treffen),
- Koordination und Unterstützung der Arbeitskreise der FG, u.a. Organisation von gemeinsamen Treffen der Fachgruppe und der Arbeitskreise und
- Kommunikationsforum der FG :
  - Pflege der www-Seiten, der email-Liste und der Mitgliederliste
  - Organisation der Diskussion über die oose-email-liste
  - Zusammenstellen einer elektronischen news-letter, regelmäßig jeden oder jeden zweiten Monat

Die einzelnen FGL-Mitglieder sollten jeweils auch für bestimmte konkrete Aufgabenbereiche verantwortlich sein.

Zusammenfassungen der Vorträge

Methoden und Werkzeuge zur Entwicklung von Systemfamilien

Matthias Riebisch, Detlef Streitferdt, Kai Böllert (Technische Universität Ilmenau)

Ein aufkommender Ansatz zur schnelleren und kostengünstigeren Softwareentwicklung sind Systemfamilien. Eine Systemfamilie beschreibt eine Gruppe ähnlicher Software-Systeme, denen eine gemeinsame Architektur sowie gemeinsam genutzte Komponenten zugrundeliegen. Darüber hinaus implementiert jedes Mitglied der Systemfamilie nur noch seine spezifischen Besonderheiten. Durch den hohen Grad an Wiederverwendung sowohl auf Modell- als auch auf Implementierungsebene können neue Familienmitglieder in kurzer Zeit mit hoher Qualität von kleinen Teams entwickelt werden.

Um erfolgreich Systemfamilien zu entwickeln, ist eine geänderte Vorgehensweise notwendig. So sind in der Analysephase nicht wie bisher die Anforderungen an ein einzelnes System zu analysieren, sondern es müssen die Gemeinsamkeiten und Unterschiede der jetzt und in der Zukunft zu erstellenden Mitglieder der Systemfamilie herausgearbeitet werden. Ein Verfahren hierfür ist “Feature-Oriented Domain Analysis” [1], welches gemeinsame und variable Eigenschaften von Familienmitgliedern in einem Feature Modell dokumentiert. Diese Eigenschaften fließen in den Architekturentwurf der Systemfamilie ein. Ferner wirken sie sich auf das Design und die Implementierung der einzelnen Komponenten hinsichtlich deren Parametrisierbarkeit aus.

Schließlich ist die Konfiguration jedes Familienmitglieds aufzustellen. Diese beschreibt die variablen Eigenschaften der einzelnen Systeme. Ausgehend von einer solchen Konfiguration baut ein Generator die konkreten Systeme zusammen [2].

Das Ziel unserer Arbeit ist die Aufstellung einer durchgängigen Methode zur Entwicklung von Systemfamilien. Diese Methode soll bestehende Verfahren integrieren und in Industrieprojekten empirisch erprobt und verfeinert werden. Begleitend entsteht eine Reihe von Werkzeugen.

Neben einer Einführung in das Thema Systemfamilien gibt unser Vortrag einen Einblick in die von uns zur Zeit bearbeiteten Fragestellungen und berichtet über erste Ergebnisse:

- Ableitung von Gemeinsamkeiten und Unterschieden aus unstrukturierten Anforderungsdokumenten
- Nutzung der UML zur Modellierung von Systemfamilien, insbesondere Integration von Feature Modellen

- Konfigurationsverwaltung der Familienmitglieder
- Automatisierte Generierung konkreter Systeme

[1] K. Kang et al. Feature-Oriented Domain Analysis (FODA) Feasibility Study. Technical Report CMU/SEI-90-TR-21, Carnegie Mellon Software Engineering Institute, 1990.
[2] D. Batory und B. Geraci. Composition Validation and Subjectivity in GenVoca Generators. IEEE Transactions on Software Engineering, Februar 1997, Seite 67-82.

Modellierung verteilter Systeme nach dem OCoN/UML-Ansatz

Holger Giese, Jörg Graf, Guido Wirtz (Universität Münster)

Die ingenieurmäßige Planung komplexer verteilter Systeme stellt die heutige Softwaretechnik einschließlich der objektorientierten immer noch vor große Probleme. Zwar scheint mit der Unified-Modeling-Language (UML), zumindest bei der Beschreibung solcher Systeme, ein wichtiger Fortschritt erreicht worden zu sein, jedoch liefern ihre Konstrukte und Beschreibungstechniken genügend Stoff zu berechtigter Kritik. So ist weder ihre informale (und damit beliebig interpretierbare) Semantik auf Dauer akzeptabel noch das völlige Fehlen einer konsistenten Beschreibungstechnik für Systemverhalten. Tragen die Konstrukte der UML zur Strukturmodellierung einerseits dazu bei, daß komplexe Systemstrukturen, ausgehend von aufgelösten Feinstrukturen bis hin zu "Big Pictures" von Systemarchitekturen relativ nahtlos und konsistent visuell beschrieben werden können, fristen ihre Konzepte zur Verhaltensmodellierung andererseits diesbezüglich ein Schattendasein. Und das obwohl tragfähige Systemstrukturen häufig erst im Kräftefeld zwischen modellierter Systemstruktur und modelliertem Systemverhalten entstehen. Die Konstrukte, die die UML zur Verhaltensmodellierung anbietet, sind entweder von fehlenden Abstraktions und Kompositionsmechanismen gekennzeichnet (vgl. Sequenz und Kollaborationsdiagramme) oder sie besitzen solche Mechanismen, diese sind aber zu denen der objektorientierter Systemmodellierung inkompatibel (vgl. Statechart- und Aktivitätsdiagramme). Ein weiterer Aspekt ist die völlige Isolation dieser Konzepte zueinander und auch zu den Konzepten der Strukturmodellierung.

Diese grundlegenden Probleme im Sprachdesign der UML führen dazu, daß der Weg zu einer nahtlosen und konsistenten Beschreibung von Systemverhalten bisher verwehrt bleibt.

Dies wird noch verstärkt, wenn bei der Beschreibung des Systemverhaltens typische Charakteristika heutiger verteilter Systeme wie z.B. Nebenläufigkeit, Verteilung, Fehlertoleranz und Zeitabhängigkeit zusätzlich berücksichtigt werden sollen. Daher ist es aus unserer Sicht unabdingbar, die Konzepte der UML zur Verhaltensbeschreibung so zu verändern, daß sie sowohl objektorientierten Abstraktions- und Kompositionsmechanismen als auch komplexen Verhaltensaspekten gleichermaßen gerecht werden. Mit dem von uns entwickelten OCoN/UML-Ansatz, der vorgestellt wird, versuchen wir diesbezüglich einen Beitrag zu leisten. Die OCoN-Sprache in diesem Ansatz repräsentiert einen Sprachformalismus, mit dem komplexes Kontroll- und Ablaufverhalten verteilter Objektsysteme größtenteils visuell beschrieben werden kann. Er besitzt einerseits eine nahtlose Einbettung in die Sprachkonzepte der UML zur Strukturmodellierung und andererseits eine formal definierte Semantik, die auf High-Level Petri-Netzen beruht.

Agentensystem zur Unterstützung von Geschäftsvorfällen im Versicherungswesen

Cay Kaben, Silke Seehusen (Fachhochschule Lübeck), Jens Seehusen (innovas, Hamburg)

Es wird ein Agentensystem vorgestellt, das die Bearbeitung von Geschäftsvorfällen innerhalb von Krankenversicherungen unterstützt. Das Agentensystem unterbreitet den Sachbearbeiterinnen Vorschläge für deren weiteres Vorgehen bei der Fallbearbeitung und weist auf das Verstreichen von Fristen hin. Die Basis für die Vorschläge bilden Regeln, die entweder vom Agentensystem automatisch generiert werden oder direkt angegeben werden können. Im laufenden Betrieb paßt das Agentensystem die Regelbasis dynamisch an.

Die einzelnen Agenten wurden objektorientiert entworfen und in Java implementiert. Die Anbindung des Agentensystems an die bestehende Anwendung erfolgt mittels Adapter-Agenten.

Die Verwaltung der Agenten übernimmt ein zentraler System-Agent, bei dem sich alle Agenten registrieren und ihre Dienste somit systemglobal anbieten. Die Kommunikation der Agenten untereinander erfolgt über Sockets.

Das Agentensystem ist mit Selbstdiagnosefunktionalität ausgestattet, die beim Erkennen von schweren Fehlern einzelne Agenten oder sogar das gesamte System deaktivieren kann.

Die Implementierung erfolgte in Java (JDK 1.2), um Plattformunabhängigkeit zu gewährleisten und die unterstützte Anwendung ebenfalls in Java implementiert wurde. Für die Darstellung und Anwendung der Regeln kommt Jess zum Einsatz.

In dem Beitrag wird gezeigt, daß eine bestehende Anwendung durch ein Agentensystem unterstützt werden kann, auch wenn dies beim Entwurf der Anwendung nicht berücksichtigt worden ist. Der objektorientierte Entwurf der Agenten ermöglicht eine einfache Erweiterung des Agentensystems um neue Agententypen.

XML Schemata – Notwendigkeit, Ansätze, Standards

Mario Jeckle (DaimlerChrysler Forschungszentrum Ulm)

Die standardisierte Auszeichnungssprache Extensible Markup Language (Abk. XML) hat mittlerweile große Bedeutung im Umfeld systemneutraler Datendarstellung erlangt. Technisch gesehen bildet XML eine Untermenge der international normierten Standardized Generalized Markup Language (Abk. SGML). Sie ist somit, wie SGML selbst, eine Metasprache – eine Sprache zur Beschreibung von Sprachen. Darüberhinaus erlaubt XML auch die Darstellung der mit den entstandenen Sprachen ausgedrückten Information. Diese Sprachen bilden die Familie der XML-Sprachen.

Vielfach konnten mit Hilfe von XML proprietäre Formate abgelöst werden. Für viele Anwendungsgebiete sind bereits XML-Sprachen vorgeschlagen, die in der Industrie große Beachtung gefunden haben. Bekannteste Beispiele dürften XHTML (der vieldiskutierte HTML-Nachfolger) für die Präsentation textueller Information, sowie die diversen Graphikformate für Pixel- und Vektorgraphiken, sowie das OMG XML Metadata Interchange Format (Abk. XMI) zur Darstellung beliebiger objektorientierter Modelle sein. Die Gründe hierfür sind sicherlich neben der leichten Lesbarkeit des vollständig in Unicode gehaltenen XML-Datenstroms auch in der Ausdrucksmächtigkeit von XML zur Strukturierung des Inhaltes zu sehen. Verglichen mit SGML rührt diese weniger von der Vielfalt der angebotenen Grundprimitive her, sondern fußt vielmehr auf deren Einfachheit was nicht zuletzt zur Verbreitung von XML durch die resultierende leichte Erlernbarkeit beitragen haben dürfte.

Zum Einsatz der angebotenen Strukturierungselemente für die Beschreibung konkreter XML-Sprache existieren verschiedene Ansätze. Zunächst die von SGML bekannten Document Type Definitions (Abk. DTD). Ihre offenkundigen Schwächen in der Beschreibung spezieller semantischer Zusammenhänge, wie sie insbesondere in der Informations- und Datenmodellierung auftreten, sollen mit die Ablösung durch Schemasprachen, die selbst in XML beschrieben sind, behoben werden. Der Vortrag gibt einen Überblick der existierenden Ansätze zur Beschreibung von XML-Schemata, sowie eine Einordnung und Bewertung relevanter Standardisierungsbemühungen.