Deutsche Forschungsgruppen in der Softwaretechnik

UniBw München

Universität der Bundeswehr München
Institut für Softwaretechnologie
AG Programmierung in speziellen Anwendungsgebieten

Prof. Dr. Uwe M. Borghoff

1 Arbeitsschwerpunkte

Die Forschungsschwerpunkte dieser Arbeitsgruppe lassen sich den beiden Gebieten "Rechnergestützte Gruppenarbeit" (engl. Computer-Supported Cooperative Work) sowie "Wissensmanagement" (speziell verteiltes Informationsmanagement) zuordnen.

Das Interesse an Anwendungen, die synchrone oder asynchrone Zusammenarbeit über das Internet ermöglichen, steigt seit geraumer Zeit stetig an. War zunächst im Kontext der sogenannten rechnergestützten Gruppenarbeit das Hauptaugenmerk auf Arbeitsgruppen gerichtet, die gemeinsam an einem Problem arbeiten, dabei aber möglicherweise räumlich und zeitlich verteilt sind, richtet sich nunmehr das (Forschungs-)Interesse in erster Linie auf die (Rechner-)Unterstützung von Communities, also auf Gruppen von Personen, die abermals räumlich und zeitlich verteilt sein können, aber kein gemeinsames Problem bearbeiten, sondern lediglich gemeinsame Interessen aufweisen.

Das Fehlen eines gemeinsamen Problems und damit verbunden eines gemeinsamen Ziels verhindert die Verwendung reiner Groupware-Anwendungen, die häufig nur aufgrund der Arbeitsgruppenzugehörigkeit nutzbringend eingesetzt werden können bzw. nur im Rahmen entsprechender Projekte akzeptiert werden. Der implizite Zwang, sich auf eine gemeinsame Arbeitsumgebung zu einigen, und die notwendigen Kompromisse zwischen Präferenzen der einzelnen Arbeitsgruppenmitglieder und Notwendigkeiten zum Funktionieren der Arbeitsgruppengesamtheit, können bei Communities nicht ohne weiteres aufrechterhalten werden.

Reine Groupware-Anwendungen können Communities nicht ohne weiteres "übergestülpt" werden. Community-Mitglieder arbeiten autonom. Hier fehlt der äußere, zielbedingte Druck und der individuelle Wille, sich solchen Zwängen zu beugen und unnötige Kompromisse zu schließen. Dennoch können Communities von einer Rechnerunterstützung profitieren. Aus diesem Grund haben sich Anfang 1999 Mitglieder zweier Münchner Universitäten (UniBWM und TUM) in der sogenannten IMC-Gruppe (Information Management for Communities) zusammengeschlossen, um den aktuellen Stand der Community-Unterstützung zu erforschen und neuartige Technologien in einschlägigen Anwendungen zu erproben.

2 Projekte

- In einem ersten Projekt betrachten wir die notwendige Infrastruktur zur Community-Unterstützung. Um den Zwang auf die Community-Mitglieder zu verringern und unnötige Kompromisse zwischen den Mitgliedern zu vermeiden, basiert unser Vorgehen auf der Autonomie der Mitglieder. Wir wählen hierzu einen agentenbasierten Ansatz. Speziell widmen wir uns Community-Agenten und Awareness-Diensten.
- Das zweite Projekt beschäftigt sich gezielt mit Cachingfragen in einer solchen Umgebung und berücksichtigt ferner Fragestellungen sogenannter lernender Benutzerschnittstellen. Dieses Projekt läuft in enger Kooperation mit dem Xerox Research Centre Europe, Grenoble.
- Ferner beschäftigt sich die Arbeitsgruppe im Rahmen eines von der Deutschen Forschungsgemeinschaft geförderten Projekts mit Fragestellungen des Informationsmanagement; hier wird in Kooperation mit der Bayerischen Staatsbibliothek im Kontext von elektronischen Bibliotheken untersucht, wie eine effiziente Langzeit-Archivierung elektronischer Dokumente durchgeführt werden kann.

3 Veröffentlichungen

1. Borghoff, U. M., Schlichter, J. H. Rechnergestützte Gruppenarbeit: Eine Einführung in verteilte Anwendungen. Springer-Lehrbuch. Berlin, Heidelberg, New York: Springer-Verlag, 2., vollst. überarb. und erw. Aufl.1998. ISBN 3-540-62873-8
2. Borghoff, U. M., Pareschi, R. (eds.) Information Technology for Knowledge Management. Berlin, Heidelberg, New York: Springer-Verlag 1998. ISBN 3-540-63764-8

3. Chidlovskii, B., Borghoff, U. M. Semantic Caching of Web Queries. The International Journal on Very Large Data Bases (to appear).

4. Borghoff, U. M., Pareschi, R., Arcelli, F., Formato, F. Constraint-Based Protocols for Distributed Problem Solving. Science of Computer Programming 30, 201-225 (1998).

5. Andreoli, J.-M., Borghoff, U. M. Virtual Answers for Query Refinement in Information Retrieval. Proc. 2nd Int'l. Conf. on the Practical Application of Constraint Technologies and Logic Programming (PACLP'2000). Blackpool, U. K.: The Practical Application Company Ltd (to appear)

6. Glance, N., Grasso, A., Borghoff, U. M., Snowdon, D., Willamowski, J. Supporting Collaborative Information Activities in Networked Communities. In: Bullinger, H.-J., Ziegler, J. (eds.): Proc. 8th Int'l. Conf. on Human-Computer Interaction (HCI'99), Vol. 2. Mahwah, NJ: Laurence Erlbaum Associates, Inc., 422-426

7. Chidlovskii, B., Borghoff, U. M. Signature File Methods for Semantic Query Caching. In: Nikolaou, C., Stephanidis, C. (eds.): Proc. 2nd European Conf. On Research and Advanced Technology for Digital Libraries (ECDL '98). Lecture Notes in Computer Science 1513. Berlin, Heidelberg, New York: Springer-Verlag, 479-498

8. Chidlovskii, B., Borghoff, U. M. Query Translation for Distributed Information Gathering on the Web. In: Eaglestone, B., Desai, B. C., Shao, J. (eds.): Proc. 2nd IEEE Int'l. Database Engineering and Applications Symposium (IDEAS'98). Los Alamitos, CA: IEEE Computer Society Press, 214-223

Kontaktadresse:

Prof. Dr. Uwe M. Borghoff
Institut für Softwaretechnologie
Universität der Bundeswehr München
85577 Neubiberg
Tel.: (089) 6004-2274/-3352/-2263
email: borghoff@informatik.unibw-muenchen.de
URL: http://ist.unibw-muenchen.de/borghoff/

UniBw München

Universität der Bundeswehr München
Institut für Softwaretechnologie
AG Grundlagen des Software-Engineering

Prof. Dr. Gunther Schmidt

1 Arbeitsschwerpunkte

Ausgehend vom Gebiet der Programmiersprachen werden Semantik von Programmiersprachen, Methoden der Programmgenerierung und Programmierkalküle studiert und dabei u.a. folgende Gebiete berührt:

Programmiermethodik und Programmtransformation

Zunehmend möchte man Programme auf einer sehr hohen Abstraktionsebene konzipieren. Sie können dann noch in mathematischer Strenge modifiziert und manipuliert werden, ehe sie durch Generatoren ihre Ausprägung in einer höheren oder niederen Programmiersprache erfahren. Von den theoretischen Grundlagen her erarbeiten wir semantische Konzepte für die Erstellung solcher Software und erproben sie modellmäßig. Praktisch umgesetzt werden rechnergestützte Erzeugung von Software und transformierende Entwicklung von Programmen aus (präalgorithmischen) Spezifikationen mit Hilfe des Programmtransformationssystems HOPS.

Relationale Methoden in der Informatik

Ausgehend von der Erkenntnis, dass zunehmender Abstraktionsgrad in Spezifikation und Komposition sich auch in der mathematischen Modellierung niederschlagen muss, propagieren wir vermehrten Einsatz relationaler Methoden in allen Bereichen der Informatik. Die Kompaktheit der Notation, die im Vergleich zu prädikatenlogischen Ansätzen höhere Abstraktionsebene, sowie die intuitivere Zugänglichkeit voll formaler Beweise prädestinieren relationenalgebraische Ansätze für Formalisierungsbestrebungen in weiten Bereichen der Softwaretechnologie. Auch im Kontext der Programmiermethodik ergeben sich immer wieder diskrete kombinatorische und graphentheoretische Probleme, für deren Bearbeitung wir relationale Methoden entwickeln.

Visuelle Compiler-Compiler

Die Vorgänge bei der Generierung und beim Einsatz von Compilern bleiben dem Anwender für gewöhnlich verborgen. Vor allem für die Ausbildung ist es daher nützlich, visuelle Compiler-Compiler ähnlich Debuggern zu verwenden, welche ihre internen Informationen visualisieren und die Abläufe so weit wie möglich durch Benutzer zu steuern gestatten. Besonders wichtig ist uns die ergonomische Gestaltung der Mensch-Maschine-Schnittstellen von Compiler-Compilern sowohl für Demonstrationmodelle als auch für Produktionssysteme: Moderne Compilertechnologie wird heute nicht nur in ihrem angestammten Bereich eingesetzt, sondern gewinnt auch als Basiswerkzeug in Bereichen der netzgestützten Dokumenten- und Datenverarbeitung, wie z.B. XML und EDI, zunehmend an Bedeutung.

2 Projekte

RelMiCS-Initiative

Im Januar 1994 konnte auf eigene Idee und mit Unterstützung der Deutschen Forschungsgemeinschaft ein einwöchiges "International Seminar on Relational Methods in Computer Science" in Schloss Dagstuhl mit 38 Wissenschaftlern aus allen Kontinenten und aus durchaus verschiedenen Disziplinen veranstaltet werden. In gemeinsamer Anstrengung wurde ein Buch (nicht nur ein Proceedings-Band) erarbeitet. Diesem folgten bereits 2 weitere Bände aus den Folgetreffen in Parati bei Rio de Janeiro (Sept. 1995) und in Hammamet (Tunesien, Jan. 1997); weitere Treffen fanden bisher in Warschau (Sept. 1998) und Québec (Jan. 2000) statt.

Aus dieser Bündelung relationaler Interessen heraus sind bereits einige spezialisiertere Initiativen hervorgegangen, unter denen wir besonders an der Organisation von Workshops im Bereich relationaler Methoden im Software-Engineering und an internationalen Kooperationen zur expliziten Verwendung relationenalgebraischer Methoden in Bereichen der Wissensrepräsentation beteiligt sind.

Programmierkalküle

Wir setzen ein ursprünglich von der DFG gefördertes Projekt zur graphisch-interaktiven Programmierung mit höheren Objekten fort; ein Schwerpunkt dieser Arbeiten ist die Erstellung des generischen Rahmenwerkes HOPS für typisierte Kalküle mit Variablenbindung. Zur Zeit arbeiten wir zudem an der Schaffung eines allgemeinen Interoperabilitätsrahmens, der die Kooperation unterschiedlicher Programmierkalküle auf dem Meta-Level zu formalisieren gestattet.

3 Veröffentlichungen

1. C. Brink, W. Kahl, G. Schmidt (eds.). Relational Methods in Computer Science. Advances in Computing Science. Springer-Verlag, Wien, New York, 1997. ISBN 3-211-82971-7.

2. B. Demuth, H. Hussmann, L. Schmitz, S. Zschaler. Erfahrungen mit einem frame workbasierten Softwarepraktikum. In Tagungsband des 6. Workshops Software-Engineering im Unterricht der Hochschulen. B. G. Teubner, Stuttgart, 1999.

3. Frank Derichsweiler. Strategiegesteuerte Programmtransformation im Higher Object Programming System HOPS. In GI, Informatiktage 1999. Konradin Verlag, 1999.

4. I. Düntsch, G. Schmidt, M. Winter. A Necessary Relation Algebra for Mereotopology. Studia Logica, 1-18 (1999).

5. A. Joaua, G. Schmidt (eds.). Special Issue on the 3rd Int. Seminar on Relational Methods in Computer Science. Information Sciences 119 (3/4) (1999).

6. W. Kahl. Relational Treatment of Term Graphs With Bound Variables. Journal of the IGPL 6 (2), 259-303 (1998).

7. W. Kahl. Internally Typed Second-Order Term Graphs. In WG '98, LNCS 1517, S. 149-163. Springer-Verlag, 1998.

8. W. Kahl. The Term Graph Programming System HOPS. In R. Berghammer, Y. Lakhnech (eds.), Tool Support for System Specification, Development and Verification, Advances in Computing Science, S. 136-149. Springer, Wien, 1999.

9. W. Kahl. Relational Matching for Graphical Calculi of Relations. Information Sciences 119 (3-4), 253-273 (1999).

10. Lothar Schmitz. Syntaxbasierte Programmierwerkzeuge. B. G. Teubner, Stuttgart, 1995.

4 Produkte und Werkzeuge

HOPS:

Das "Higher Object Programming System" HOPS ist ein graphisch-interaktives Programmier- und Programmtransformationssystem, das die Programmstruktur in der Form von streng typisierten Termgraphen direkt zugänglich macht. HOPS ist ausgelegt für transformationelle Programmentwicklung, bei der das Software-Endprodukt ausgehend von abstrakten Spezifikationen und Algorithmenformulierungen durch semantikerhaltende Transformationsschritte hergeleitet wird, mit dem Ziel der korrekten Programmentwicklung. Weitere Informationen unter der URL:

http://ist.unibw-muenchen.de/HOPS/

Jaccie:

Aufbauend auf dem mit dem Deutschen Hochschul-Software-Preis der ASK ausgezeichneten Vorsystem

SIC enthält Jaccie einen Werkzeugsatz mit Scannergenerator, verschiedenen Parsergeneratoren, Editoren für Grammatiken und Attributierungen, interaktiven Debugging-Werkzeugen für alle Phasen, Informationsfenstern für Grammatikinfos und Automaten in Klartextdarstellung. Auf Knopfdruck entstehen Java-Quelltexte von Scannern und Parsern, die in andere Anwendungen eingebunden werden können. Weitere Informationen unter der URL:

http://ist.unibw-muenchen.de/Tools/JACCIE/

SalesPoint Framework (mit TU Dresden):

Es ist nicht einfach, Studenten vor dem Vordiplom in einer Veranstaltung Teamarbeit, Prinzipien systematischer Software-Entwicklung, neue Sprachen, Entwicklungs- und Dokumentationswerkzeuge nahezubringen. Die Grundidee, als gemeinsame Basis für die Studentenprojekte ein Framework (das die Erstellung von Verkaufsanwendungen unterstützt) zu verwenden, bringt eine Reihe von Vorteilen mit sich:

- Studenten lernen, sich mit bestehender Software auseinanderzusetzen.
- Mit dem Framework ist auch der OO-Entwurf der zu erstellenden Anwendung vorgegeben und damit eine Hauptschwierigkeit für Anfänger entschärft.
- Es lassen sich relativ leicht neue und vergleichbare Aufgaben finden.

  Das umfassend dokumentierte Framework stellt aufeinander abgestimmte Komponenten bereit wie Kataloge und Bestände, Oberflächenbausteine zur Warenauswahl und Katalogpflege, sowie Datenkörbe zur Modellierung von Verkaufsprozessen mit Transaktionsunterstützung. Weitere Informationen finden Sie unter der URL:

  http://ist.unibw-muenchen.de/Lectures/SalesPoint

Kontaktadresse:

Prof. Dr. Gunther Schmidt
Institut für Softwaretechnologie
Universität der Bundeswehr München
85577 Neubiberg
Tel.: (089) 6004-2249/-2263
email: schmidt@informatik.unibw-muenchen.de
URL: http://ist.unibw-muenchen.de/schmidt/

UniBw München

Universität der Bundeswehr München
Institut für Softwaretechnologie
AG Informationssysteme

Prof. Dr. Andy Schürr

1 Arbeitsschwerpunkte

Die Forschungsaktivitäten der Arbeitsgruppe lassen sich alle dem Gebiet "Definition und Einsatz visueller Sprachen in der Softwaretechnik" zuordnen. Im einzelnen handelt es sich dabei um:

OO-Modellierungssprachen und -Werkzeuge:

Die von der OMG als Standard verabschiedete OO-Modellierungssprache UML ist ein Hauptgegenstand unserer aktuellen Forschungsaktivitäten. Zum einen befassen wir uns mit der präzisen Definition der Semantik verschiedener UML-Teilsprachen, zum anderen steht die Anpassung von UML an das Anwendungsgebiet der eingebetteten Echtzeitsysteme und ihr Einsatz zur Modellierung von Softwarearchitekturen im Vordergrund.

Visuelle graphbasierte Spezifikationssprachen:

Graphtransformationssysteme lassen sich der Kategorie der regelbasierten visuellen Sprachen zuordnen. Kombiniert mit objektorientierten und imperativen Konzepten werden sie von uns als ausführbare Spezifikationssprache mit einer präzsen Semantikdefinition sowie als VHL-Programmiersprache für die Generierung von "CASE Tools" eingesetzt.

Formale Definition visueller Sprachen:

Heute verbreitete visuelle Modellierungs- und Programmiersprachen besitzen in aller Regel weder eine präzise Syntax- noch eine vollständige Semantikdefinition. Wir untersuchen deshalb den Einsatz von Graphgrammatiken zur Syntaxdefinition visueller Sprachen und entwickeln hierzu entsprechende Parsing-Algorithmen. Darüber hinaus werden für die Semantikdefinition solcher Sprachen nichtmonotone Logiken, Fixpunkttheorie und ggf. auch Kategorientheorie eingesetzt.

Konstruktion verteilter Anwendungen:

Bei der Konstruktion verteilter Anwendungssysteme spielen neben den bekannten Middleware-Technologien (CORBA, DCOM, -) auf Architekturebene sogenannte "Design Patterns" eine große Rolle. Basierend auf UML als Architekturbeschreibungssprache widmen wir uns hier vor allem der halbautomatischen Verteilung objektbasierter monolithischer Software. Dabei werden Graphtransformationsregeln zur Identifikation sogenannter "Anti-

Patterns" in einer gegebenen Softwarearchitektur und zur Elimination identifizierter "Anti-Patterns" durch die Verteilung unterstützende "Design Patterns" eingesetzt.

2 Projekte

Automotive UML (UML-Erweiterung)

Gemeinsam mit der DaimlerChrysler AG (Standort Esslingen) und der AG Informationssysteme der Universität Paderborn entwickeln wir ein sogenanntes UML-Profil (Erweiterung, Spezialisierung) für die Konstruktion von eingebetteter Echtzeitsoftware im Automobilbau.

PROGRES (Programmieren mit Graphen)

Gemeinsam mit dem Lehrstuhl für Informatik III der RWTH Aachen entwickeln wir eine integrierte Umgebung, die das Programmieren mit UML-Klassendiagrammen, Graphtransformationsregeln und imperativen Kontrollstrukturen unterstützt und als "Meta-CASE-Tool" für visuelle Modellierungswerkzeuge eingesetzt wird.

Regelbasierte Generierung von Testszenarien

Gemeinsam mit der DaimlerChrysler AG (Standort Berlin) entwickeln wir ein regelbasiertes und frei konfigurierbares Transformationssystem, das Klassifikationsbäume (für Eingabedaten) aus Software-Modellen extrahiert und als Eingabe für die eigentliche Testvorbereitung und Planung ergänzt und aufbereitet.

APPLIGRAPH / GETGRATS

Im Rahmen dieser beiden EU-Projekte werden Aktivitäten zur Entwicklung und Anwendung graphtransformationsbasierter Spezifikationssprachen und -werkzeuge unterstützt. Mit unseren PROGRES-Aktivitäten sind wir hier vor allem im Bereich Anwendung und Werkzeugbau aktiv.

3 Veröffentlichungen

1. A. Schürr: Programmed Graph Replacement Systems, in: G. Rozenberg (ed.): Handbook on Graph Grammars: Foundations, Vol. 1, Singapur: World Scientific (1997), 479-546

2. J. Rekers, A. Schürr: Defining and Parsing Visual Languages with Layered Graph Grammars, in: Journal of Visual Languages and Computing, Vol. 8, No. 1, London: Academic Press (1997), 27-55

3. P. Klein, A. Schürr: Constructing SDEs with the IPSEN Meta Environment, in: Proc. 8th Conf. on Software Engineering Environments SEE'97, Los Alamitos: IEEE Computer Society Press (1997), 2-10

4. D. Blostein, A. Schürr: Computing with Graphs and Graph Rewriting, in: Software - Practice Experience, Vol. 29, No. 3, Los Alamitos: IEEE Computer Society Press (1999), 1-21

5. A. Schürr, A.J. Winter: UML, the Future Standard Software Architecture Description Language?, in: H. Kilov, B. Rumpe, I. Simmonds (eds.): Behavioral Specifications for Businesses and Systems, Kluwer (1999), 193-206

6. A. Schürr, A. Winter, A. Zündorf: PROGRES: Language and Environment, in: H. Ehrig, G. Engels, H.-J. Kreowski, G. Rozenberg (eds.): Handbook on Graph Grammars: Applications, Languages, and Tools, Vol. 2, Singapur: World Scientific (1999), 487-550

7. A. Radermacher: Support for Design Patterns through Graph Transformation Tools, erscheint in: M. Nagl, A. Schürr, M. Münch (eds.): Proc. AGTIVE Workshop, LNCS, Springer Verlag (2000)

4 Produkte und Werkzeuge

PROGRES (für Linux):

Bei PROGRES handelt es sich um eine integrierte Programmierumgebung mit Grafikeditoren und Texteditoren für Klassendiagramme, Graphersetzungsregeln, Attributierungsgleichungen und imperative Kontrollstrukturen sowie mit Analysatoren, Interpreter, Compiler und tcl/tk-basierter Rapid-Prototyping-Generator für grafische Editoren. Weitere Informationen unter der URL:

http://ist.unibw-muenchen.de/Tools/PROGRES

DiTo (Distributed Transformation Tool):

DiTo ist ein mit PROGRES und JavaCC teilweise generiertes Werkzeug zur Transformation monolithischer Java-Programme in verteilte CORBA-basierte Anwendungen. Transformationen werden auf Architekturebene (UML-Klassendiagrammen) und auf Quelltextebene simultan durchgeführt. In einem "Nachlauf" werden middleware-spezifische Codestücke generiert und eingefügt. Weitere Informationen unter der URL:

http://ist.unibw-muenchen.de/Tools/dito

Kontaktadresse:

Prof. Dr. Andy Schürr
Institut für Softwaretechnologie
Universität der Bundeswehr München
85577 Neubiberg
Tel.: (089) 6004 2220, Fax: (089) 6004 3560
email: Andy.Schuerr@unibw-muenchen.de
URL: http://ist.unibw-muenchen.de/schuerr