Workflow- und Prozeßsynchronisation mit Interaktionsausdrücken und -graphen: Konzeption und Realisierung eines Formalismus zur Spezifikation und Implementierung von Synchronisationsbedingungen

Christian Heinlein
heinlein@informatik.uni-ulm.de

Abt. Datenbanken und Informationssysteme
Fakultät für Informatik
Universität Ulm
D-89069 Ulm

Betreuer der Arbeit: Prof. Dr. Peter Dadam
Art der Arbeit: Dissertation zur Erlangung des Doktorgrades Dr. rer. nat.
Fachbereich GI: 2 (Softwaretechnologie und Informationssysteme)

Zusammenfassung

Im Rahmen der Arbeit werden *Interaktionsausdrücke* und *-graphen* als *deskriptiver Formalismus* zur kompakten, übersichtlichen und modularen *Spezifikation* sowie zur effizienten *Implementierung* von Synchronisationsbedingungen unterschiedlichster Art konzipiert, theoretisch untersucht, praktisch implementiert und prototypisch zur Synchronisation von Workflows angewandt.

1. Problemstellung

Heutige Workflow-Management-Systeme – egal ob kommerzielle Produkte oder Forschungsprototypen – betrachten einzelne Workflows im wesentlichen als separate, in sich abgeschlossene Prozesse, die *unabhängig* voneinander ausgeführt werden. In der Regel werden weder Beschreibungsmittel zur Spezifikation noch Laufzeitmechanismen zur Überwachung von *workflowübergreifenden Synchronisationsbedingungen* angeboten, obwohl derartige Bedingungen in vielen Anwendungsbereichen eine wichtige Rolle spielen. Auch auf den ersten Blick naheliegende Lösungsansätze, wie z. B. das Verschmelzen abhängiger Workflows oder eine explizite nachrichten- oder regelbasierte Synchronisation von Workflows, erweisen sich aus verschiedenen Gründen als nicht gangbare Wege. Zum einen würden verschmolzene Workflows in der Praxis Größenordnungen annehmen, die nicht mehr handhabbar sind, zum anderen müßten sämtliche möglichen Kombinationen von parallel ausführbaren Workflows vormodelliert werden. Außerdem sind die genannten Ansätze prinzipiell nicht in der Lage, *dynamisch veränderliche* Workflow-Geflechte zu beschreiben, bei denen weder die Anzahl noch die Typen der parallel ausgeführten Workflows a priori bekannt sind. Ziel der vorliegenden Arbeit [4] war daher die Konzeption und Realisierung eines Formalismus, mit dessen Hilfe derartige *Inter-Workflow-Abhängigkeiten* sinnvoll beschrieben und implementiert werden können.

2. Überblick

Im Rahmen der Arbeit werden *Interaktionsausdrücke* und *-graphen* als *deskriptiver Formalismus* zur kompakten, übersichtlichen und modularen *Spezifikation* sowie zur effizienten *Implementierung* von Synchronisationsbedingungen unterschiedlichster Art konzipiert, theoretisch untersucht, praktisch implementiert und prototypisch zur Synchronisation von Workflows angewandt.

Ausgehend von einer ausführlichen, anwendungsorientierten und anschaulichen Beschreibung des Formalismus, werden sukzessive eine präzise *formale Semantik*, eine äquivalente *operationale Semantik* sowie eine effiziente *Implementierung* entwickelt. Aufbauend auf der formalen bzw. operationalen Semantik werden zahlreiche *formale Eigenschaften* sowie Aussagen zur *Ausdrucksmächtigkeit* und *Komplexität* des Formalismus bewiesen. Darüber hinaus wird aufgezeigt, wie Interaktionsausdrücke unter anderem zur Spezifikation von *Inter-Workflow-Abhängigkeiten* eingesetzt werden können und wie die spezifizierten Bedingungen mit Hilfe eines oder mehrerer *Interaktionsmanager* in eine Workflow-Ausführungsumgebung integriert werden können. Um die Erstellung von Interaktionsgraphen und ihre Transformation in äquivalente Interaktionsausdrücke zu erleichtern, wurde außerdem ein syntaxgesteuerter graphischer *Editor* entwickelt.

Im einzelnen konnten die in den folgenden Abschnitten 3 bis 6 skizzierten Resultate erzielt werden.

3. Entwicklung von Interaktionsgraphen

Ausgehend von regulären Ausdrücken mit Operatoren für sequentielle Komposition (Sequenz), sequentielle Iteration (Kleenesche Hülle) und Disjunktion (Auswahl), konnte durch die Hinzunahme „komplementärer" bzw. dualer Operatoren für parallele Komposition, parallele Iteration und Konjunktion sowie durch die Einführung parametrisierter Ausdrücke und Quantoren ein Formalismus zur Spezifikation von Synchronisationsbedingungen entwickelt werden, der auf der einen Seite konzeptionell einfach und übersichtlich ist und auf der anderen Seite eine große Flexibilität und Ausdrucksmächtigkeit besitzt. Dementsprechend kann er erfolgreich zur Lösung einer großen Palette unterschiedlichster Synchronisationsprobleme eingesetzt werden, wenn man die von einem Ausdruck akzeptierten Worte als *zulässige Ausführungsreihenfolgen* von Aktionen interpretiert. Durch die intuitive graphische Notation, die es erlaubt, diese Aktionsfolgen anschaulich durch das Traversieren des zugehörigen Graphen zu ermitteln, sowie durch den Einsatz benutzerdefinierter Operatoren lassen sich auch komplizierte Ausdrücke anwenderfreundlich darstellen.

4. Formale Semantik und Eigenschaften

Durch eine geeignete Verallgemeinerung des „klassischen" sprachtheoretischen Ansatzes, wie er beispielsweise zur Definition der Sprache eines regulären Ausdrucks eingesetzt wird, ist es gelungen, eine formale Semantik für Interaktionsausdrücke zu entwickeln, die das anschauliche Prinzip der Graphtraversierung – auch für Spezialfälle wie Sackgassen und endlose Wege – geeignet präzisiert. Aufbauend auf dieser Semantik konnten zahlreiche intuitiv einleuchtende Eigenschaften von Interaktionsausdrücken, wie z. B. Kommutativität, Assoziativität oder Idempotenz bestimmter Operatoren, auch formal verifiziert werden. Außerdem wurden Interaktionsausdrücke bzgl. ihrer Ausdrucksmächtigkeit mit regulären Ausdrücken und kontextfreien Grammatiken verglichen.

5. Operationale Semantik, Implementierung und Komplexität

Trotz zum Teil erheblicher Schwierigkeiten, insbesondere im Kontext parametrisierter Ausdrücke und Quantoren, ist es gelungen, ein vollständiges operationales Modell zur Ausführung von Interaktionsausdrücken zu entwickeln und seine Korrektheit in bezug auf die formale Semantik nachzuweisen. Unter Zuhilfenahme einer geeigneten Programmiersprache konnte dieses Modell von Zuständen, Zustandsübergängen und Zustandsprädikaten erfolgreich in eine kompakte und effiziente Implementierung von Interaktionsausdrücken überführt werden.

Obwohl es prinzipiell (wie erwartet) Interaktionsausdrücke gibt, für die die Komplexität des Wort- oder Aktionsproblems[1] – bei Verwendung der vorliegenden Implementierung – exponentiell bzgl. der Länge der Eingabe ist, konnten dennoch einige wichtige Aussagen formuliert und bewiesen werden, mit deren Hilfe für alle bisher betrachteten praktisch relevanten Ausdrücke gezeigt werden kann, daß sie gutartig sind, d. h. mit polynomieller Komplexität verarbeitet werden können.

6. Anwendung

Durch die Definition geeigneter Koordinationsprotokolle konnte die zunächst isoliert entwickelte Implementierung von Interaktionsausdrücken mit einer Schnittstelle umgeben werden, die es anderen Programmen erlaubt, zuvor spezifizierte Integritätsbedingungen bei der Ausführung von Aktivitäten tatsächlich zu berücksichtigen. Anhand zweier alternativer Ansätze zur Implementierung von Workflow-Geflechten – Adaption von Arbeitslistenprogrammen einerseits und Adaption von Workflow-Ausführungseinheiten andererseits – wurde aufgezeigt, wie die genannten Koordinationsprotokolle praktisch angewandt werden können. Um Interaktionsgraphen auch in großen verteilten Systemen sinnvoll und zuverlässig einsetzen zu können, wurden Konzepte zur Partitionierung von Graphen, zum Einsatz mehrerer Interaktionsmanager und zum Wiederanlauf nach Systemausfällen erarbeitet. Desweiteren wurde ein einfaches Vorgehensmodell zur Definition von Workflow-Geflechten entwickelt.

---

[1] Im Gegensatz zum klassischen Wortproblem besteht das Aktionsproblem darin, ausgehend von den bereits ausgeführten Aktionen zu entscheiden, ob eine gegebene Aktion momentan zulässig ist oder nicht.

7. Verwandte Arbeiten

Beim Vergleich von Interaktionsausdrücken mit anderen auf regulären Ausdrücken basierenden Formalismen zur Spezifikation von Synchronisationsbedingungen – etwa Pfadausdrücken [2], Synchronisierungsausdrücken [3] oder Ereignis- und Flußausdrücken [10, 11] – stellt man fest, daß alle verwandten Ansätze mehr oder weniger lückenhaft sind. Beispielsweise unterstützen Synchronisierungsausdrücke zwar die parallele Komposition von Teilausdrücken, nicht jedoch die zugehörige parallele Iteration, während es sich bei Pfadausdrücken gerade umgekehrt verhält. Ereignis- und Flußausdrücken wiederum fehlt das für praktische Anwendungen wesentliche Konzept parametrisierter Ausdrücke und Quantoren. Außerdem leiden fast alle verwandten Formalismen an mangelnder Orthogonalität, da bei der Kombination der angebotenen Operatoren zum Teil erhebliche Einschränkungen beachtet werden müssen. Beide Mängel – konzeptionelle Unvollständigkeit und fehlende Orthogonalität – wurden bei der Entwicklung von Interaktionsausdrücken vermieden.

Beim Vergleich von Interaktionsausdrücken mit Formalismen wie Prozeßalgebren [z. B. 5, 6, 9] oder Petrinetzen [1, 8] fällt auf, daß letztere zwar sehr ausdrucksstark sind, ihr zugrundeliegendes Ausführungsmodell aber grundsätzlich nichtdeterministische Entscheidungen zuläßt, was für das primäre Anwendungsgebiet von Interaktionsausdrücken – Synchronisation paralleler Workflows – nicht akzeptabel ist. Außerdem werden wichtige Entwurfsziele von Interaktionsausdrücken, wie z. B. die modulare Kombinierbarkeit von Ausdrücken oder die Erweiterbarkeit des Formalismus um benutzerdefinierte Operatoren, nicht zufriedenstellend unterstützt.

Obwohl das Problem der Inter-Workflow-Abhängigkeiten in vielen Anwendungen von Workflow-Management von praktischer Bedeutung ist, wird es in der „Workflow community“ nach wie vor „stiefmütterlich“ behandelt. Beispielsweise findet man weder in entsprechenden Themenheften von Zeitschriften [z. B. 13–15] noch in Büchern, die den Anspruch erheben, den „State of the art“ widerzuspiegeln [7, 12], Beiträge, die sich wirklich mit diesem Thema auseinandersetzen. Auch in Konferenzbänden und Workshop-Proceedings, in denen das Thema Workflow-Management an sich immer mehr an Bedeutung gewinnt, sucht man i. d. R. vergeblich nach einschlägigen Artikeln. Somit stellt die Entwicklung und der Einsatz von Interaktionsausdrücken und -graphen zur Koordination dynamisch veränderlicher Workflow-Geflechte einen wichtigen Beitrag zur Lösung des Inter-Workflow-Koordinationsproblems dar.

Referenzen

[1] B. Baumgarten: *Petri-Netze. Grundlagen und Anwendungen* (2. Auflage). Spektrum Akademischer Verlag, Heidelberg, 1996.

[2] R. H. Campbell, A. N. Habermann: „The Specification of Process Synchronization by Path Expressions.“ In: E. Gelenbe, C. Kaiser (eds.): *Operating Systems* (International Symposium; Rocquencourt, France, April 1974; Proceedings). Lecture Notes in Computer Science 16, Springer-Verlag, Berlin, 1974, 89–102.

[3] L. Guo, K. Salomaa, S. Yu: „On Synchronization Languages.“ *Fundamenta Informaticae* 25 (3+4) March 1996, 423–436.

[4] C. Heinlein: *Workflow- und Prozeßsynchronisation mit Interaktionsausdrücken und -graphen. Konzeption und Realisierung eines Formalismus zur Spezifikation und Implementierung von Synchronisationsbedingungen*. Dissertation, Fakultät für Informatik, Universität Ulm, 2000.

[5] M. Hennessy: *Algebraic Theory of Processes*. The MIT Press, Cambridge, MA, 1988.

[6] C. A. R. Hoare: *Communicating Sequential Processes*. Prentice-Hall, London, 1985.

[7] S. Jablonski, M. Böhm, W. Schulze (eds.): *Workflow-Management. Entwicklung von Anwendungen und Systemen*. dpunkt-Verlag, Heidelberg, 1997.

[8] K. Jensen, G. Rozenberg (eds.): *High-Level Petri Nets. Theory and Application*. Springer-Verlag, 1991.

[9] R. Milner: *Communication and Concurrency*. Prentice-Hall, New York, 1989.

[10] W. E. Riddle: „A Method for the Description and Analysis of Complex Software Systems.“ *ACM SIGPLAN Notices* 8 (9) September 1973, 133–136.

[11] A. C. Shaw: „Software Description with Flow Expressions.“ *IEEE Transactions on Software Engineering* SE-4 (3) May 1978, 242–254.

[12] G. Vossen, J. Becker (eds.): *Geschäftsprozeßmodellierung und Workflow-Management. Modelle, Methoden, Werkzeuge*. International Thomson Publishing, Bonn, 1996.

[13] Special Issue on Workflow Systems. *Distributed Systems Engineering Journal* 3 (4) December 1996.

[14] Themenheft Workflow-Management. *Informatik Forschung und Entwicklung* 12 (2) May 1997.

[15] Special Issue on Workflow and Process Management. *Journal of Intelligent Information Systems. Integrating Artificial Intelligence and Database Technologies* 10 (2) March 1998.