

Reihe 20

Rechnerunter-
stützte Verfahren

Nr. 469

Dipl.-Ing. Matthias Jüttner,
Mosbach

Softwareagenten- basierte Berechnung interdisziplinärer gekoppelter Simulationen



Softwareagenten-basierte Berechnung interdisziplinärer gekoppelter Simulationen

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart zur Erlangung der Würde eines
Doktor Ingenieurs (Dr.-Ing.) genehmigte Abhandlung

vorgelegt von

Matthias Johannes Helmut Jüttner

aus Mosbach

Hauptberichter: Prof. Dr. techn. Wolfgang M. Rucker

Mitberichter: Prof. Dr.-Ing. Dr. h. c. Peter Göhner

Prof. Dr. sc. techn. Jan Hesselbarth

Tag der mündlichen Prüfung: 13. November 2017

Institut für Theorie der Elektrotechnik

Universität Stuttgart

2017

Fortschritt-Berichte VDI

Reihe 20

Rechnerunterstützte
Verfahren

Dipl.-Ing. Matthias Jüttner,
Mosbach

Nr. 469

Softwareagenten-basierte
Berechnung
interdisziplinärer
gekoppelter Simulationen

VDI verlag

Jüttner, Matthias

Softwareagenten-basierte Berechnung interdisziplinärer gekoppelter Simulationen

Fortschr.-Ber. VDI Reihe 20 Nr. 469. Düsseldorf: VDI Verlag 2017.

124 Seiten, 49 Bilder, 7 Tabellen.

ISBN 978-3-18-346920-8, ISSN 0178-9473,

€ 48,00/VDI-Mitgliederpreis € 43,20.

Für die Dokumentation: kooperative Feldberechnung – Softwareagenten – gekoppelte Simulation – intelligente Systeme – paralleles Rechnen

Der wachsende Bedarf an interdisziplinären und gekoppelten physikalischen Simulationen auf verteilten und sich dynamisch verändernden Rechnernetzen motiviert den Einsatz des hier vorgestellten Softwareagentensystems. Die Umsetzung entspricht einer neuen Abstraktionsebene oberhalb bestehender Feldberechnungswerkzeuge und ermöglicht deren dynamisches, automatisiertes und leistungsbezogenes Zusammenwirken. Ressourcenabhängig spezialisierte Berechnungseinheiten agieren dabei autonom. Im Kollektiv ermöglichen sie die Beantwortung von Fragen, zu der Einzelne nicht fähig sind. Das System ist einfach erweiterbar und unterstützt Anwender durch intelligente Algorithmen und Erfahrungswissen z. B. bei der Konfiguration des Lösungsverlaufs und bei der Reduktion von Rechenzeit. Die kooperative Feldberechnung gelingt mittels dargestellter Schnittstellen. Herausfordernde Simulationsbeispiele belegen den vorteilhaften Einsatz dieses Systems und den so gewonnenen Mehrwert.

Bibliographische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet unter <http://dnb.ddb.de> abrufbar.

Bibliographic information published by the Deutsche Bibliothek

(German National Library)

The Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliographie (German National Bibliography); detailed bibliographic data is available via Internet at <http://dnb.ddb.de>.

D 93

© VDI Verlag GmbH · Düsseldorf 2017

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen oder vollständigen Wiedergabe (Fotokopie, Mikrokopie), der Speicherung in Datenverarbeitungsanlagen, im Internet und das der Übersetzung, vorbehalten.

Als Manuskript gedruckt. Printed in Germany.

ISSN 0178-9473

ISBN 978-3-18-346920-8

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Theorie der Elektrotechnik der Universität Stuttgart.

Mein besonderer Dank geht an Herrn Prof. Dr. techn. Wolfgang M. Rucker für das Ermöglichen dieser Arbeit, seine vielseitige Unterstützung im Verlauf der Arbeit sowie die Übernahme des Hauptberichts.

Mein Dank gebührt ebenso Herrn Prof. Dr.-Ing. Dr. h. c. Peter Göhner für die vielfältigen Anregungen während des Entstehens dieser Arbeit sowie die Übernahme des Mitberichts.

Herrn Prof. Dr. sc. techn. Jan Hesselbarth sei für das Ermöglichen der abschließenden Arbeiten und den Mitbericht gedankt.

Allen Kolleginnen und Kollegen gilt mein Dank für die gute Zusammenarbeit, die Unterstützung sowie die vielen hilfreichen Diskussionen.

Ebenfalls sei allen Studierenden gedankt, die im Rahmen ihrer studentischen Tätigkeit zum Gelingen der Arbeit beigetragen haben und mich durch ihr Wirken in der Lehre unterstützten.

Abschließend geht mein Dank an die Familie und die vielen Freunde, die mich mit ihrer Fürsorge stets begleitet haben.

„So ist's ja besser zu
zweien als allein.“

(Kohélet 4,9)

Inhaltsverzeichnis

Zusammenfassungen	XII
1 Einleitung	1
1.1 Motivation	1
1.2 Zielsetzung	3
1.3 Gliederung	5
2 Softwareagentensysteme	6
2.1 Agentenorientierte Programmierung	6
2.2 Softwareagenten	7
2.3 Softwareagentensysteme	9
2.4 Herausforderungen	12
2.5 JADE	13
3 Numerische Berechnung gekoppelter Feldsimulationen	15
3.1 Methode der finiten Elemente im Überblick	15
3.2 Gekoppelte Simulationen	19
3.2.1 Stark oder schwach gekoppelte Simulationen	20
3.2.2 Zeitliche Kopplungen	21
3.2.3 Raumbasierte Kopplungen	22
3.2.4 Gebietskopplungen	24
3.2.5 Methodenkopplungen	25
3.3 Gleichungssysteml�ser	25
3.3.1 Monolithische L�ser	25
3.3.2 Gestaffeltes L�sen	29
3.4 Berechnungsumgebungen	31
4 Softwareagenten-basierte Simulationsumgebung	33
4.1 Systemarchitektur	33
4.1.1 Agentenbeschreibungen	36
4.1.2 Kommunikationsarten	42
4.1.3 Ausf�hrung von Verhalten	46
4.1.4 Vererbung	47

4.2	Numerischer Lösungsprozess	48
4.2.1	Initialisierung und Segmentierung	48
4.2.2	Iterative Kopplung	50
4.2.3	Lösungsverlauf und Konvergenz	53
4.2.4	Agentenspezifischer Berechnungsschritt	60
4.2.5	Gebiets- und Methodenkopplung	62
4.2.6	Umgang mit unterschiedlicher Diskretisierung	67
4.3	Ressourcennutzung	68
4.3.1	Agenteninterne Rechenzeitnutzung	68
4.3.2	Rechenzeit des Agentensystems	73
5	Numerische Beispiele	76
5.1	Nahbereichsradar	76
5.1.1	Konkurrierende Wellensimulationen	76
5.1.2	Interdisziplinär gekoppelte Simulationen	78
5.2	Transistor	86
5.2.1	Ortsabhängige Kopplungen	87
5.2.2	Zeitabhängige Kopplungen	91
6	Resümee und Ausblick	95
	Literaturverzeichnis	98

Formelzeichen und Abkürzungen

Lateinische Buchstaben

A	Magnetisches Vektorpotenzial
a	Ordnung der Stetigkeit
B	Magnetische Flussdichte
\tilde{C}	Adjazenzmatrix einer Kopplung
C^a	Stetigkeit der a . Ordnung
D	Elektrische Flussdichte
D_n	Diffusionskoeffizient der Elektronenkonzentration
D_p	Diffusionskoeffizient der Löcherkonzentration
\mathcal{D}	Diskretisierung
E	Elektrische Feldstärke
E	Anzahl deformierter Referenzelemente
f	FEM-Energiefunktional
G	Matrix stationärer iterativer Verfahren
g	Vektor stationärer iterativer Verfahren
h	Elementgröße
I	Identitätsmatrix
I	Elektrischer Strom
J	Volumenstromdichte
\tilde{J}	Jacobimatrix
j	Komplexe Einheit
K	Systemmatrix
L	Untere Dreiecksmatrix
M	Massenmatrix
N	Fremdatomkonzentration
N_A^-	Akzeptorenkonzentration
N_D^+	Donatorenkonzentration
N_i	Formfunktion
n	Normalenvektor
n	Elektronenkonzentration
n	Frequenztransformierte Elektronenkonzentration
n_i	Intrinsische Ladungsträgerdichte
P	Permutationsmatrix
P_e	Péclet-Zahl

p	Löcherkonzentration
\mathfrak{p}	Frequenztransformierte Löcherkonzentration
Q	Orthogonale bzw. unitäre Matrix
Q	Punktladung
q	Wärmequellendichte
R	Obere Dreiecksmatrix
r	Lastvektor
S	Energieflussdichte
\hat{S}	Schurkomplement
T	Temperatur
t	Zeit
U	Untere Dreiecksmatrix
U	Elektrische Spannung
u	Lösungsvektor
\mathfrak{u}	Frequenztransformierter Lösungsvektors
W	Energie
W_E	Energie des elektrischen Feldes
W_Q	Energie einer Ladungsverteilung
X	Suchrichtungsvektor nicht stationärer iterativer Verfahren
x	Ortsvektor in kartesischen Koordinaten

Griechische Buchstaben

α	Skalierungsfaktor nicht stationärer iterativer Verfahren
Γ	Rand des Gebiets Ω
Γ_D	Rand mit Dirichlet-Randbedingungen
Γ_N	Rand mit Neumann-Randbedingungen
γ	Häufigkeitsverteilung
δ	Schwellwert zum Berechnungsabbruch
ε_0	Elektrische Feldkonstante
ε_r	Relative Permittivität
ϵ	Fehlerschranke
κ	Elektrische Leitfähigkeit
κ_t	Thermische Leitfähigkeit
Λ	Lagrange-Multiplikator
μ_r	Relative Permeabilität
μ_0	Magnetische Feldkonstante
μ_n	Elektronenbeweglichkeit
μ_p	Löcherbeweglichkeit
ν	Testfunktion

ρ	Volumenladungsdichte
σ	Flächenladungsdichte
τ	Periode eines harmonischen Signals
ϕ	Frequenztransformiertes elektrisches Potenzial
φ	Elektrisches Potenzial
φ_i	Ansatzkoeffizienten des elektrischen Potentials
Ω	Gebiet
$\bar{\Omega}$	Gebiet außerhalb von Ω
ω	Betrag der Winkelgeschwindigkeit

Symbole und Operatoren

div	Divergenz
grad	Gradient
*	Komplexe Konjugation
Δ	Laplace-Operator
∂	Partielle Ableitung
\Re	Realteil einer komplexen Zahl
\top	Transponierte Matrix
\cup	Vereinigungsmenge
\cap	Schnittmenge
\approx	Approximation
\emptyset	Leere Menge
\circ	Hadamard-Produkt
$ $	Betrag
$ $	Norm

Abkürzungen

ACG	Agentenbasiertes Rechencluster (Agent-Based Computational Grid)
AMS	Agent Management System
API	Schnittstelle zur Anwendungsprogrammierung
BA	Berechnungsagent
BDDC	Gebietszerlegungsmethode (Balancing Domain Decomposition by Constraints)
BDI	Wissen-Wünsche-Ziele-Verhaltensmodell (Belief-Desires-Intentions behaviour model)
BEM	Randelementmethode
BiCGStab	Lösungsverfahren (Bi-Conjugate Gradient Stabilized Method)
Bitkom	Bundesverband Informationswirtschaft, Telekommunikation und neue Medien e.V.
BOINC	Rechencluster (Berkeley Open Infrastructure for Network Computing)
CG	Lösungsverfahren (Conjugate Gradient Method)
CORBA	Anwendungsarchitektur zur Vermittlung von Objekt-Nachrichten (Common Object Request Broker Architecture)
CPU	Hauptprozessor (Central Processing Unit)
CT	Containertabelle (Container Table)
DA	Dokumentationsagent
DD	Gebietszerlegungsverfahren (Domain Decomposition Method)
DF	Softwareagent (Directory Facilitator)
EMV	Elektromagnetische Verträglichkeit
FEM	Finite Elemente Methode
FETI	Gebietszerlegungsverfahren (Finite Element Tearing and Interconnecting)
FETI-DP	Gebietszerlegungsverfahren (FETI Dual-Primal)
FGMRES	Lösungsverfahren (Flexibel Generalized Minimal Residual Method)
FIPA	Gesellschaft für intelligente physikalische Agenten (Foundation for Intelligent Physical Agent)
FLOPS	Gleitkommaoperationen pro Sekunde (Floating Point Operations Per Second)
GADT	Beschreibung global ausgeführter Agenten (Global Agent Descriptor Table)
GPI	Programmierschnittstelle zur globalen Speicheradressierung (Global Address Space Programming Interface)
GMRES	Lösungsverfahren (Generalized Minimal Residual Method)
JADE	Agentenentwicklungsumgebung (Java Agent DEvelopment Framework)
JVM	Virtuelle Javamaschine (Java Virtual Machine)
KA	Koordinationsagent
KI	Künstliche Intelligenz
LADT	Beschreibung lokal ausgeführter Agenten (Local Agent Descriptor Table)
LGS	Lineares Gleichungssystem

M2M	Maschine zu Maschine (Machine to Machine)
MG	Mehrgitterlösungsverfahren (Multigrid)
MpCCI	Diskretisierungsabhängige Schnittstelle zur Quelltextkopplung (Mesh-based parallel Code Coupling Interface)
MPI	Schnittstelle zum Nachrichtenaustausch (Message Passing Interface)
MUMPS	Lösungsverfahren (MULTifrontal Massively Parallel sparse direct Solver)
OpenMP	Schnittstelle zur parallelen Programmierung (Open Multi-Processing)
PARDISO	Lösungsverfahren (Parallel Sparse Direct and Multi-Recursive Iterative Linear Solvers)
PDE	Partielle Differenzialgleichung (Partial Differential Equation)
RMI	Aufruf entfernter Methoden (Remote Method Invocation)
RPC	Vereinheitlichter Funktions- und Methodenaufruf durch verteilte Systeme (Remote Procedure Call)
SOAP	Protokoll zum Datenaustausch und für Funktionasaufrufe (Simple Object Access Protocol)
SOR	Überrelaxationsverfahren (Successive Over-Relaxation)
SORV	Überrelaxationsverfahren vektoriell
VDE	Verband der Elektrotechnik, Elektronik und Informationstechnik
VDI	Verein Deutscher Ingenieure
VDMA	Verband Deutscher Maschinen- und Anlagenbau e.V.
XML	Erweiterbare Auszeichnungssprache (Extensible Markup Language)
ZVEI	Zentralverband Elektrotechnik- und Elektronikindustrie e.V.

Zusammenfassung

Die Notwendigkeit der Simulation interdisziplinärer und gekoppelter physikalischer Zusammenhänge auf verteilten und sich dynamisch verändernden Rechnernetzen motiviert den Einsatz des im Rahmen dieser Arbeit entwickelten und im Folgenden vorgestellten Softwareagentensystems. Dabei entsprechen interdisziplinär gekoppelte Systeme solchen, deren Berechnung durch ein Zusammenwirken mehrerer spezialisierter Einheiten gelingt. Beispiele stellen verschiedene sich beeinflussende physikalische Zusammenhänge dar, die sich verschiedenen Ingenieursdisziplinen zuordnen lassen. Weitere Beispiele sind sich gegenseitig beeinflussende Teilgebiete oder Zeitabhängigkeiten in einer Simulation. Die Umsetzung des Softwareagentensystems entspricht dabei der Einführung einer neuen Abstraktionsebene oberhalb bestehender Simulationswerkzeuge, die deren Zusammenwirken ermöglicht. Dazu notwendig werde eine Beschreibung der Eigenschaften und Fähigkeiten der gekapselten Simulationswerkzeuge sowie die Umsetzung von Schnittstellen zur numerischen Zusammenarbeit. Die entstehenden spezialisierten Berechnungseinheiten nutzen zur Simulation die ihnen verfügbare Soft- und Hardware und reagieren selbstständig auf bereitgestellte Zwischenergebnisse. Das so realisierte neuartige System ermöglicht es unterschiedlichen und unabhängigen Simulationseinheiten automatisiert zusammenzuarbeiten.

Die kooperative Feldberechnung gelingt durch das Bereitstellen von Randbedingungen und Quellen während der Feldberechnung sowie durch das Auslesen von Ergebnissen an beliebigen Stellen des Modells. Erforderliche Funktionen der Schnittstellen, beispielsweise zur Steuerung und Überwachung von Feldberechnungssoftware, werden hier dargestellt. Jedem Berechnungsagenten wird es so möglich, die globale Simulationaufgabe entsprechend seiner Vorstellungen eines zielführenden Lösungsbeitrags zu modifizieren. Die entstehenden Teilaufgaben werden dann angepasst auf ihre jeweiligen Anforderungen berechnet, was zu einer schnellen Lösungsbereitstellung führt.

Das vorgestellte Agentensystem ermöglicht die kollektive Bearbeitung von Aufgaben, zu der ein Einzelner nicht fähig ist. Zudem ist das dynamisch entstehende Simulationssystem einfach um Fähigkeiten und Ressourcen erweiterbar. Ein gezieltes Hinzufügen von zusätzlichen Ressourcen mit gleichen Fähigkeiten ermöglicht eine Varianz und Redundanz im Lösungsverlauf. Dabei werden existierende Teilergebnisse umgehend berücksichtigt, sodass das Agentensystem automatisiert leistungsbezogenen Lösungswege vergleicht. Dies gestattet die eigenständige Konfiguration des Lösungsablaufs durch beteiligte Berechnungsagenten und den automatischen

Einsatz der zur Berechnung notwendigen Simulationswerkzeuge. Zusätzliche Ressourcen dienen dem Aufbau von Erfahrungswissen, das hier mittels eines lernfähigen Systems verwaltet und zur Reduktion benötigter Rechenzeit verwendet wird.

Bei numerischer Betrachtung entspricht der mittels Softwareagenten umgesetzte Lösungsverlauf einer gestaffelten Berechnung, die für viele gekoppelte numerische Simulationen zielführend ist. Das Einbeziehen von Agenten, deren Berechnungsfähigkeiten zur alleinigen monolithischen Berechnung ausreichen, ermöglicht eine zusätzliche konkurrierende Berechnung sowie weitere Entscheidungsoptionen in Bezug auf die Durchführung der Berechnung. Um die Autonomie der Berechnungsagenten hervorzuheben, wird auf eine zentrale Instanz während der Berechnung verzichtet. Die detaillierte Darstellung des durch Kooperation erreichten numerischen Lösungsprozesses verdeutlicht zusätzlich die Funktionsweise des Agentensystems.

Die Berechnung herausfordernd ausgewählter Simulationsbeispiele belegt, dass die kooperative Berechnung interdisziplinärer gekoppelter Simulationen durch autonome Berechnungseinheiten in verteilten Systemen auch für verschiedene Diskretisierungen gelingt. Zudem veranschaulichen die Beispiele den durch die agentenbasierte Herangehensweise gewonnenen Mehrwert, wie beispielsweise seine Flexibilität und Zuverlässigkeit. Im Hinblick auf die Verwaltung der zur Simulation notwendigen Infrastruktur gelingt die automatisierte Nutzung von auf die betrachtete Simulation angepassten und befähigten Ressourcen. Die durchgeführte Rechenzeitauswertung und die Betrachtung der Ressourcennutzung demonstrieren die Leistungsfähigkeit des realisierten Systems für unterschiedliche Kopplungen.

Das vorgestellte Agentensystem unterstützt dabei Anwender, die Simulationen zur Produktgestaltung verwenden. Beispiele dafür stellen die Konfiguration der Randbedingungen zur Kopplung von Simulationen, die dynamische Anpassung der Lösungssequenz an verfügbare Rechenressourcen oder die intelligente Entscheidung der Softwareagenten für ein Lösungsverfahren dar. Entsprechend ermöglicht es eine Reduktion der zur Durchführung einer Simulation erforderlichen menschlichen Arbeitszeit. Zudem unterstützt es selbst erfahrene Anwender bei getroffenen Entscheidungen bzw. bestätigt diese darin.

Abstract

Simulating interdisciplinary and coupled equations on distributed and dynamically changing computer networks motivates the usage of a software agent system, developed and presented within this thesis. These interdisciplinary and coupled systems mean that a cooperative calculation is possible by multiple specialized units. Examples are different physics that correspond to various engineering disciplines and have an effect on each other. Other examples are different parts of a simulation model or time steps within a simulation model influencing each other. The implementation of the software agent system adds a new abstraction layer above existing simulation tools and enables its cooperation. It requires a description of the features and capabilities of the encapsulated simulation tools as well as the implementation of interfaces used for numeric cooperation. The resulting specialized calculation units' uses own hard- and software for calculation and react independently to provide partial results. The resulting novel system enables individual and independent simulation units to do cooperative work automatically.

The cooperative field calculation get possible by providing boundary conditions and coupling sources during the field calculation and extracting results at arbitrary coordinates within the simulation model. Required interface functionality, for example to the control and monitor the field calculation software, are shown here. Using these, each calculation agent gets capable of modifying the global simulation task, according to its ideas of a target-oriented solution contribution. This allows a calculation of the resulting partial tasks with respect to their needs and consequently a fast deployment of the solutions.

The introduced software agent system enables a cooperative treatment of tasks, which an individual agent is not capable of handling. In addition, the dynamically jointed simulation system is simple to extend by abilities and resources. Pointedly added additional resources with equal capabilities add variance and redundancy in the solution process. Once partial results are provided, they are taken into account by the software agent system. This enables an automatic comparison of the performance of all applied solutions strategies. It allows the involved calculation agents to alter the solution process and to apply the required simulation tools. Additional resources increase the experiential knowledge, managed here by a learning system that reduces calculation time.

The numerical analysis of the software agents based solution process corresponds to a segregated calculation method, suitable for most coupled numerical simulations. Involving agents with sufficient calculation capabilities to monolithically simulate the model add an additional compe-

titive solution strategy as well as another option for calculation. To emphasise the autonomy of the calculation agent a central instance during calculation is forewent. The detailed description of the numerical solution process achieved by the cooperating software agent system additionally highlights its workflow.

The calculation of the selected challenging simulation examples shows that the cooperative calculation of interdisciplinary and coupled simulations works for autonomous calculation units on distributed systems, even for different discretizations. The examples also show the additional benefit obtained by the software agent based approach, such as its flexibility and reliability. Even the management of required simulation infrastructure profits since only proper and versed resources are automatically used for the calculation of the considered simulation. The evaluation of the computation time and the considered resources usage demonstrates the performance of the implemented system for different couplings.

The presented software agent system supports users doing simulations for product design. Examples are the setup of boundary conditions for coupled simulations, the dynamic adjustment of the solution process to available computing resources or the intelligent decision of software agents for solution methods. Accordingly, it allows a reduction of required human working time. It also supports and encourage even trained users while solving simulations.

