

Розробка засобів обчислювальної техніки та дослідження комп'ютерних мереж

V.A. Svyatnyy¹,
V. G. Kuschnarenko^{1,3},
M. Resch²,
O.M. Miroschkin¹,
S. Wesner³,

¹Donezker Nationale Technische Universität, Ukraine

²Höchstleistungsrechenzentrum Stuttgart, Stuttgart

³Institut für Organisation und Management von Informationssystemen Universität Ulm,

vsvjatnyj@gmail.com, resch@hlrs.de, stefan.wesner@uni-ulm.de,

volodymyr.kuschnarenko@uni-ulm.de

Problematik der parallelen Simulationstechnik

Die parallele Simulationstechnik wird im Rahmen einer Kooperation von Informatikern und Simulationstechnikern der Universität Stuttgart, der Universität Ulm und der Donezker Nationalen Technischen Universität (Pokrowsk) als ein aktuelles Forschungsgebiet untersucht. Im Mittelpunkt stehen dabei neue parallele numerische Verfahren, die theoretische Analyse von virtuellen parallelen Simulationsmodellen und ihre Devirtualisierung sowie die Entwicklung einer leistungsfähigen und benutzerfreundlichen Simulationsumgebung mit einer effizienten Nutzung von parallelen Hard- und Softwareressourcen. Mit dem entwickelten Simulationswerkzeug werden komplexe Modelle von dynamischen Systemen mit konzentrierten und mit verteilten Parametern numerisch gelöst. Als Beispielprozesse werden Modelle von Grubenbewetterungssystemen und aus der Verfahrenstechnik in entsprechenden problemorientierten verteilten parallelen Simulationsumgebungen (VPSU) betrachtet.

Einführung

Die Anwendungsgebiete der Simulationstechnik reichen von den "traditionellen" technischen dynamischen Systemen (DS) mit konzentrierten und mit verteilten Parametern (DSKP, DSVP) über biotechnologische und ökologische Prozesse bis hin zu den zu projektierenden und zu optimierenden Unternehmens-, Wirtschafts- und Öffentlichkeitsstrukturen. Die modellgestützte Entwicklung von komplexen technischen und nichttechnischen DS in allen Bereichen ist ein entscheidender Faktor zur Qualitätssicherung und Konkurrenzfähigkeit bei laufenden und zukünftigen technischen und technologischen Projekten. Als Folge von steigenden Anforderungen an die Qualität, Sicherheit und Wirtschaftlichkeit der DS-Prozesse werden immer genauere und damit auch komplexere Simulationsmodelle benötigt. Bei dieser Entwicklung ergibt sich in einer natürlichen Weise eine Komplexitätsschwelle für die Implementierung und Anwendung der Simulationsmodelle, die nur durch die Verwendung von neuen Simulationswerkzeugen überwunden werden kann. Hierzu gehören u.a. die parallelen Höchstleistungsrechner der MIMD-Architektur mit eng (über gemeinsamen Adressraum von Shared Memory) und lose (mit dem

autonom zugeordneten Adressraum) gekoppelten Prozessoren sowie mit einer hybriden Speicherorganisation. Die effiziente Nutzung von diesen Simulationswerkzeugen erfordert eine benutzerfreundliche system- und informationstechnische Organisation der parallelen Ressourcen sowie eine hoch entwickelte Modellierungs- und Simulationssoftware, die neue Ansätze zur Modellerstellung und Simulationsdurchführung zur Verfügung stellt. Die hierbei zu lösenden theoretischen und praktischen simulationstechnischen Probleme gehören zum Fachgebiet der **parallelen Simulationstechnik (ParSimTech)**. Dieses Forschungsgebiet wird im Rahmen der Kooperation von Informatikern und Simulationstechnikern der Universität Stuttgart, der Universität Ulm und der Donezker Nationalen Technischen Universität (Pokrowsk, Ukraine) untersucht.

Dynamische Systeme als Objekte der Modellierung und Simulation

Als dynamische werden die Objekte von Technik und Technologien bezeichnet, deren Betriebsart mit den Prozessen von Zustandsänderungen zeitabhängig charakterisiert wird. Diese Prozesse können von den Naturerscheinungen oder zielgerichteten Steuerungsaktionen für die Überführung der

Objekte aus gewissen Anfangszuständen in bestimmten vordefinierten Zustand iniiert werden. Jeder technischer Bereich, jede Technologie hat spezifischen dynamischen Objekte und Prozesse, die im Rahmen von verfahrenstechnischen Fließbilder sowie in der Integration mit den Automatisierungsmitteln folgende Eigenschaften von komplexen dynamischen Systemen (KDS) bekommen: große Anzahl und Mehrfachheit der Verbindungen von charakteristischen Parametern, die Nichtlinearität der statischen und dynamischen Charakteristiken der Komponenten, hierarchische Anordnung von Regelungsmitteln, die Abhängigkeit der Prozesse von zeitlichen und örtlichen Koordinaten, die physikalische Vielfältigkeit der zusammenwirkenden Prozesse, eine Wechselung der Strukturen abhängig von Abläufen der dynamischen Prozesse, eine Vorhandenheit der Phasen und Bereiche von Parametern, die sicherheitskritische Zustände der Systeme verursachen können u. a. Wissenschaftstechnisches Niveau der jetzigen Gegenstandsgebiete (GG, Abb. 1) von der Bergbautechnik, des Hüttenwesens bis der Bioverfahrens- und Umweltschutztechnik wird in bedeutendem Ausmaß durch erreichte Wissensenebene der Prozeßtheorie, des Verständnisses der Zusammenwirkungen und der Integration verschiedener dynamischen Prozesse sowie ihren praktischen Inbetriebnahmen von GG-Experten in KDS-Industrieprojekte bestimmt. KDS-Theorie vermittelt eine Annäherung der Gegenstandsgebiete (Abb.1) bei der kompletten Lösung von technischen und technologischen Problemen.

Aus simulationstechnischen Gründen werden zwischen dynamischen Systeme mit konzentrierten (DSKP) und örtlich verteilten (DSVP) Parametern unterscheidet. Traditionell gehören die Theorie und Simulationsverfahren von DSKP und DSVP zu den selbstständigen wissenschaftlichen Disziplinen. Forschungs- und Entwicklungspraktiken zeigen aber, dass die DSKP und DSVP oft als dieselben Objekte untersucht werden: üblicherweise wird ein Objekt erst als VP-Objekt analysiert, dann werden die weitere Untersuchungen durch eine Approximation mit der gewissen Menge der KP-Objekte und der Behaltung von entsprechenden Approximationsbedingungen fortgesetzt. Eine Analyse zeigt, dass trotz breite physikalische DS-Vielfältigkeit der verschiedenen Gegenstandsgebiete ihre Topologien mit der in Abb.1 illustrierten relativ begrenzten Menge von Darstellungsmitteln beschrieben werden: das sind die verfahrenstechnische Fließbilder (VTF), Strukturschemata der Automatisierungssysteme (SSA), durch den Graphen dargestellten dynamischen Netzobjekte (DNO) und die während der Approximation von DSVP entstehenden sekundären Topologien (ST).

Hauptziel der Topologiebeschreibung besteht in eindeutlicher Darstellung der DS-Struktur, der Bestandteile und ihrer Verbindungen sowie in der Integration mit der mathematischen Beschreibung in der Form, die als Voraussetzung für automatische Generierung der großdimensionalen Prozessgleichungssysteme dienen konnte. Die Bestandteile der obigen Topologien können DSKP und DSVP sein. Die Herleitung der Prozessgleichungen gehört in alle Gegenstandsgebiete zur Forschungsproblematik, welche das neue Wissen als mathematische Modelle liefert. In der Abb.1 sind die Gleichungsarten gezeigt: die DSKP werden mit den algebraischen und gewöhnlichen differentialen Gleichungssysteme während die DSVP mit den partiellen Differentialgleichungssysteme beschrieben. Die aktuellen möglichen Untersuchungsebenen von KDS zeigt Abb.2 [6, 12].

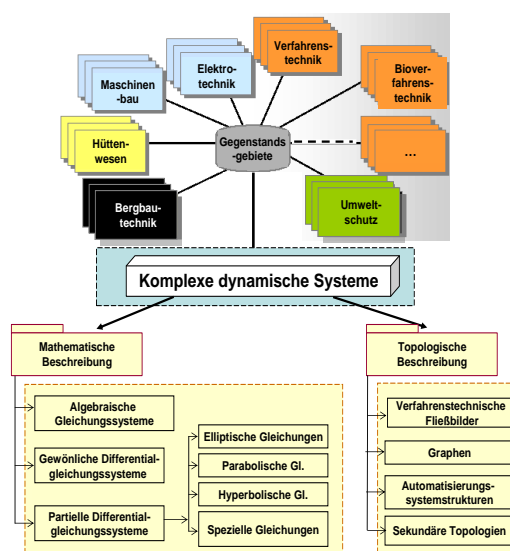


Abbildung 1. Anwendungsgebiete und Beschreibungen von komplexen dynamischen Systemen

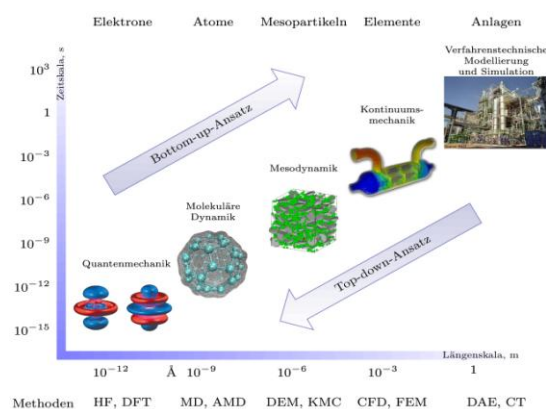


Abbildung 2. Mögliche Untersuchungsebenen von komplexen dynamischen Systemen

Die mathematische Modelle werden aufgrund folgender Methoden entwickelt: HF – Hartree-Fock-Verfahren; DFT – Density functional theory; MD – Molekularodynamik; DEM – Discrete element method; KMC – Kinetic Monte Carlo; CFD – Computational fluid dynamic; FEM – Finite element method; DAE – Differential algebraic equation; CT – Control theory.

Etappen der parallelen DSKP-, DSVP-Modellierung. Funktionalität von Simulationsmittel

Definition 1: Das Modell des komplexen dynamischen Systems ist eine formale KDS-Beschreibung, die notwendige Topologiedarstellung und entsprechende Gleichungssysteme der physikalischen Prozesse beinhaltet und durch die gemeinsamen Vektor-Matrix-Ausdrücke und -Operationen diese Mittel vereinigt (Abb. 1).

Definition 2: Das Simulationsmodell des komplexen dynamischen Systems mit konzentrierten Parametern ist das umgewandelte DSKP-Modell zur Form, die Anwendung von numerischen Lösungsverfahren erlaubt.

Definition 3: Das Simulationsmodell des komplexen dynamischen Systems mit verteilten Parametern ist das örtlich diskretisierte und umgewandelte DSVP-Modell zur Form, die Anwendung von numerischen Lösungsverfahren erlaubt.

Wegen der örtlichen Diskretisierung des DSVP entsteht eine sekundäre Topologie, die in der Modellumwandlung eine wichtige Rolle spielt. Die partielle Differentialgleichungssysteme werden durch die gewöhnliche Differentialgleichungssysteme (GDGLS) approximiert. In den DSKP- und DSVP-Simulationsmodellen werden die GDGLS bezüglich der Vektoren von Ableitungen ausgelöst. Die vorhandenen algebraischen Gleichungssysteme werden bezüglich der Vektoren der gesuchten Variablen ausgelöst.

In Abb.3 werden die Etappen der Entwicklung von Modellen der DSKP, DSVP aller genannten Topologien bis zur Umwandlung in den Simulationsmodelle dargestellt. Wegen der DS-Komplexität spielt bei der DSKP- und DSVP-Modellierung die rechnergestützte Erstellung der Topologien, Generierung der Gleichungssysteme sowie ihre Umwandlung in Simulationsmodelle als allgemeines und gegenstandsgebietspezifisches Problem der parallelen Simulationstechnik eine zunehmende Rolle. In Abb.3 wird das Vorhandensein von Mitteln für rechnergestützte Erfüllung entsprechenden Arbeiten vorausgesetzt. Erfahrung zeigt aber, dass diese Mittel in den mehreren Gegenstandsgebiete noch entwickelt wer-

den sollen. Es geht um benutzerfreundlichen simulationsorientierten Bedienoberflächen (BOF), Funktionsalgorithmen und Software der Topologieanalyatoren und Gleichungsgeneratoren.

Die erhaltenen Simulationsmodelle dienen für weitere Schritte der parallelen Simulationstechnik, die mit folgenden Definitionen verbunden sind.

Definition 4: Parallele Modellierung von dynamischen Systemen mit konzentrierten und verteilten Parametern ist ein Vorgang, der zielgerichtet zur Realisierung der obigen DSKP-, DSVP-Simulationsmodelle auf dem vorgegebenen parallelen Rechnersystem führt.

Definition 5: Virtueller MIMD-Prozess ist ein relativ autonomes Programm, das den Lösensalgorithmus des Gleichungssystemteils realisiert und mit den in dem Lösen des Gesamtgleichungssystem beteiligt Prozessen kommuniziert.

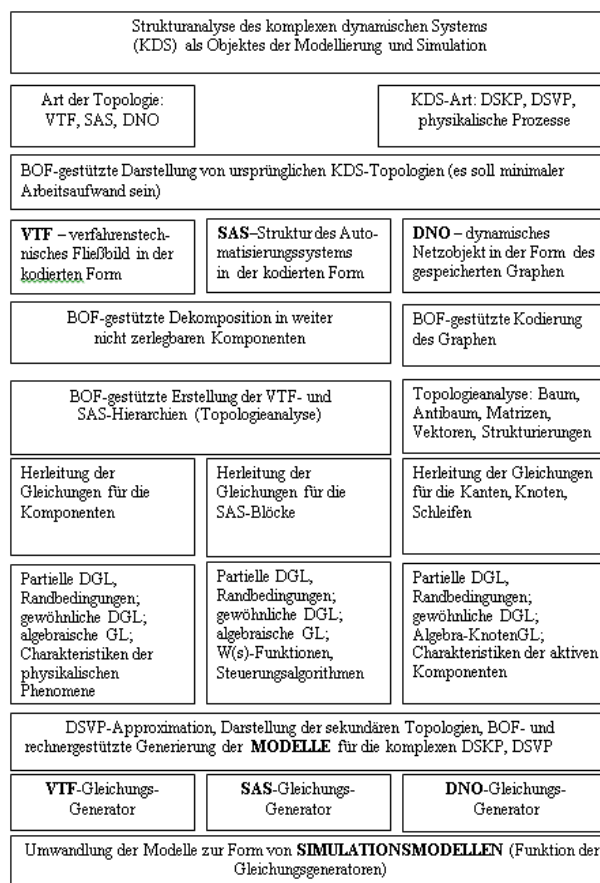


Abbildung 3. Etappen der rechnergestützten Entwicklung von DSKP- und DSVP-Simulationsmodellen

Die Körnigkeit des virtuellen Prozesses charakterisiert den Umfang der Berechnungs- und Hilfsoperationen, die im Prozess programmiert werden. Dieser Begriff ist mit der KDS-Dekomposition (Topologie und Modelle) sowie mit der örtlichen DSVP-

Approximation verbunden, weil während der Dekomposition bekommen wir die KDS-Einheiten, die in den weiter nicht zerlegbaren Elementen diskretisiert werden. Die DSVP-Ortsdiskretisierung führt zur sekundären VTF-, SAS- und DNO-Topologien, deren Elemente mit entsprechendem System der gewöhnlichen Differentialgleichungen beschrieben werden. Die Körnigkeit der virtuellen Prozesse wird nach der Betrachtung der Gleichungssysteme aller nach der Dekomposition erhaltenen VTF-, SAS- und DNO-Komponenten und Approximationselementen definiert. Diese Einheiten haben folgende Eigenschaften, die wir für die weitere Parallelisierung in den Betracht nehmen sollen:

- alle Komponenten funktionieren kontinuierlich, also parallel und asynchron zu einander;
- die Ortsdiskretisierung von Komponenten gibt die finite Elemente, die zeitabhängig kontinuierlich bleiben und funktionieren parallel;
- die Kontinuität der Elemente, Komponente und VTF, SAS und DNO insgesamt bedeutet einerseits eine inhärente Parallelität, verursacht aber andererseits eng gekoppelte Berechnungsprozesse, die auf jedem Zeitschritt einer Datenaktualisierung von Nachbarelementen, Reglern, Energie- und Störungsquellen sowie Knotenverbindungen brauchen.

Die erhaltenen Arten von Gleichungen, die die KDS-Komponente und ihre Elemente beschreiben, lassen uns die Körnigkeit von virtuellen Prozessen definieren. Dabei werden wir davon ausgehen, dass die parallelen Rechensysteme die fein- und grobkörnige formale Konstrukte parallel bearbeiten. Die feinkörnige Parallelitätsebene wird durch die einzige Operationen präsentiert, welche im SIMD-System mit Hilfe des HOST-Rechners und der Felder von Prozessorelementen realisiert werden. Die grobkörnige Parallelitätsebene wird durch einige Mengen der Operationen präsentiert und dem Prinzip der MIMD-Parallelität entspricht. Dabei ändert sich die im Begriff „Prozeß“ eingebettete Menge der Operationen sehr stark – von dem Einanweisungsprogramms bis den recht komplexen Programmen. Wir werden aus unserer Erfahrung der Aufbau von SIMD- und MIMD-Gleichungslösern ausgehen [2,4,11,19] und die Perspektive der Realisierung von grobkörnigen Prozessen in den modernen MIMD-Systeme mit der hybriden Programmiermodelle berücksichtigen. Analyse des Cauchy-Problems für gewöhnlichen Differentialgleichungen zeigt, dass es sich nicht lohnt, parallel durch Prozesse die Glieder von rechten Sei-

ten der Gleichungen zu berechnen. Als versprechend für weitere Untersuchungen sieht ein Prozess aus, der für die Berechnung der gesuchten Variablen zuständig wäre.

Definition 6 : Die **minimalkörnigen virtuellen Prozesse** beschränken sich mit den Lösensoperationen für die Gleichungen, die nach der Dekomposition und der Diskretisierung der VTF-, SAS- und DNO-Komponenten entstanden sind.

Die vorgeschlagene minimale Körnigkeit von Prozessen werden wir für die Erstellung der virtuellen parallelen Simulationsmodelle benutzen.

Definition 7: Virtuelles paralleles DSKP/DSVP-Simulationsmodell (VPSM) ist eine Abstraktion, die aus dem bezüglich den Ableitungsvektoren und den Vektoren der gesuchten algebraischen Unbekannten aufgelösten Gleichungssystem und der nach einem Parallelisierungsansatz entwickelten Struktur der virtuellen MIMD-Prozesse für paralleles Lösen der Gleichungssystemteile besteht.

Minimaler Gleichungssystemteil ist eine gewöhnliche Differentialgleichung (oder eine algebraische Gleichung), für deren Lösung ein minimalkörnigen MIMD-Prozess zuständig ist. Für die Realisierung der oben definierten virtuellen Prozesse wählen wir die numerische Verfahren, die in den konventionellen Simulationssprachen Simulink, ACSL, Dymola, Modelica u.a. verwendet werden. Für die Untersuchung sind auch die **blockartige Verfahren** mit interner Parallelität aktuell.

Die Berechnungen der Vektorelemente nach entsprechenden numerischen Verfahren bedingen den Arbeitsaufwand (die Last) von virtuellen Prozessen. Die VTF-, SAS- und DNO-Gleichungsgeneratoren berücksichtigen die ausgewählte Verfahren und generieren die diskrete virtuelle parallele Simulationsmodelle (**DVPSM**) der DSKP, DSVP als Berechnungsformeln (explizite Verfahren) oder Gleichungssysteme (implizite Verfahren) für gesuchten Variablen der minimalkörnigen Prozesse.

Definition 8: Virtuelles Verbindungsnetzwerk ist abstrakter Schaltautomat, der die logischen Verbindungen zwischen den virtuellen MIMD-Prozessen widerspiegelt.

Definition 9: Die virtuelle parallele Rechnerarchitektur ist eine über virtuellen Verbindungsnetzwerk gekoppelte, nach MIMD-Prinzip funktionierende nicht beschränkte Menge der vollfunktionellen Prozessoren, die gemeinsam unter Berücksichtigung des algorithmisch bedingten Datenaustausches das Lösen des Simulationsproblems durchführen.

Definition 10: Das Zielrechnersystem (ZRS) ist die dem Modellentwickler zur Verfügung stehende lose oder/und eng gekoppelte, nach MIMD-Prinzip funk-

tionierte, vorgegebene Menge der installierten vollfunktionellen Prozessoren mit lokalem oder/und gemeinsamem Speicher und verfügbarem vordefiniertem programmgesteuertem Verbindungsnetzwerk.

Definition 11: Paralleler DSKP-, DSVP-Simulator ist vorgegebenes Ziel der parallelen Modellierung und wird als eine Hardware/Software-Systemorganisation definiert, die den Modellierungsvorgang und diskretes virtuelles paralleles Simulationsmodell auf dem Zielrechnersystem softwaretechnisch effizient realisiert und den Modellentwicklern sowie den Modellanwendern benutzerfreundliche Kommunikation mit den Modellierungs- und Simulationsressourcen erlaubt.

Definition 12: Eine Devirtualisierung des diskreten virtuellen parallelen Simulationsmodells ist eine Umwandlung, die zur Simulatorrealisierung auf gegebenen ZRS eindeutig führt und aus folgenden Grundfunktionen besteht: Apriori-Analyse des DVPSM; Darstellung der vorhandenen ZRS-Ressourcen; Zuordnung "virtuelle Prozesse - reale Prozessoren", "virtuelles Verbindungsnetzwerk (Kommunikationsgraph) - reales ZRS-Verbindungsnetzwerk" mit der Berücksichtigung von Kriterien der Lastbalancierung, Netzwerkanpassung, Minimierung des Datenaustauschufwandes; Formierung der Spezifikation von MIMD-Prozessen, die den Prozessoren zugeordnet werden; Formierung des Auftrages für paralleler Code-Generator.

Es wird vorgeschlagen, die parallelen Simulatoren von komplexen DSKP, DSVP entsprechend den in Abb. 4 dargestellten Etappen und Vorgehen der parallelen Modellierung zu entwickeln. Ausgangspunkt der Entwicklung ist eine detaillierte Darstellung und Apriori-Analyse der parallelen virtuellen Simulationsmodelle.

Die Ziele der Apriori-Analyse sind: ausgehend von minimalkörnigen Prozessen mögliche Ansätze zur Parallelisierung und die strukturelle Darstellungen der DSKP-, DSVP- Parallelitätsebenen zu betrachten; die Parallelitätsebene nach Kriterien Lastverteilung, Datenaustausch, Synchronisation der Prozesse, Rechenaufwand zu prüfen; die möglichen alternativen Lösungen der Organisation von Parallelitätsebenen vorzuschlagen und zu prüfen; Zusammenfassender Vergleich der Parallelitätsebenen durchzuführen und die Empfehlungen für die Implementierungsarbeiten auf möglichen ZRS zu formulieren.

Die Parallelisierungsansätze werden im Zusammenhang mit der Approximation der DSVP bezüglich der Ortskoordinaten und der Dekomposition von VTF-, SAS- und DNO-Simulationsobjekten auf den weiter nicht zerlegbaren Elementen betrachtet. Die erhaltenen VPSM mit minimalkörnigen Prozes-

sen werden wir als paralleles Simulationsmodell der ersten Parallelitätsebene betrachten. Diese Ebene (Abb.4) beinhaltet maximal mögliche Menge der Einheiten „Gleichung - Prozess“ und dient für die Untersuchungen der potentiellen Parallelität von VTF-, SAS-, DNO- und sekundären Topologien. Die Komposition der zerlegten Objekte und eine Gruppierung der minimalkörnigen Prozesse führen zur Betrachtung der höheren Parallelitätsebenen mit der größeren Prozesskörnigkeit und entsprechenden Varianten der Prozess-Processor-Zuordnung.

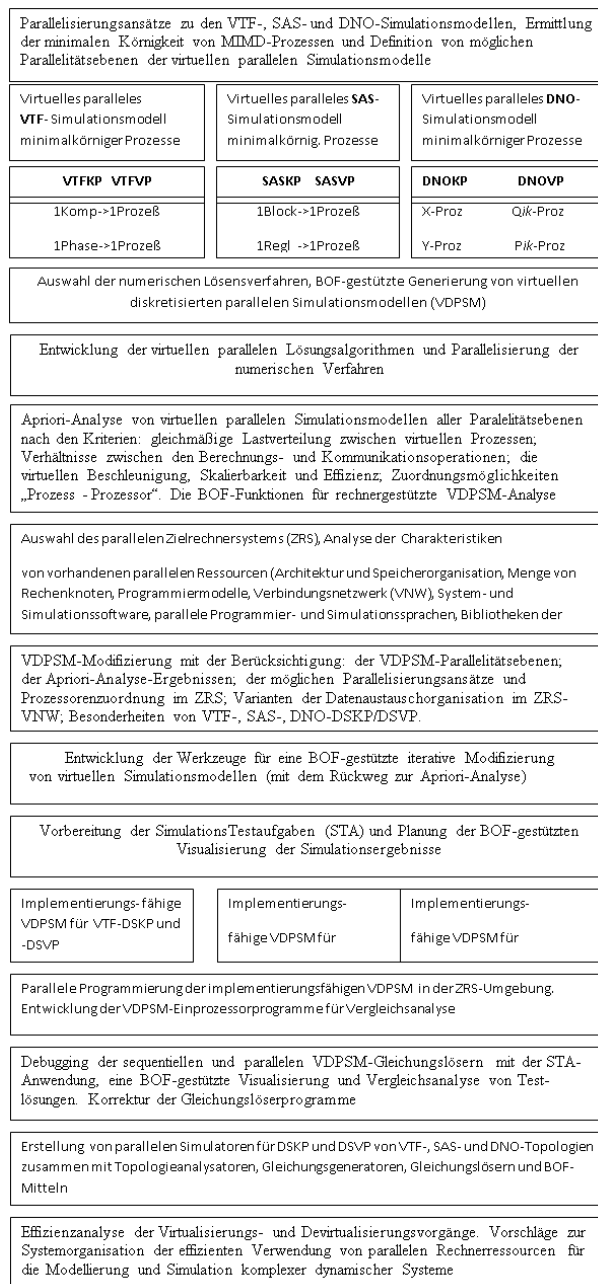


Abbildung 4. Etappen der parallelen KDS-Modellierung und rechnergestützten Entwicklung von parallelen Simulatoren

Die Apriori-Analyse soll bezüglich der aller möglichen Parallelitätsebene vollständig durchgeführt werden und in der Betrachtung folgenden Fragen ihrer Funktionierung bestehen:

- Analyse des Operationenumfanges im virtuellen Prozess;
- Vergleich der Prozesse nach Kriterium „Prozesslasthöhe“;
- Vorschläge zur möglich notwendigen Lastbalancierung;
- Analyse der Kommunikationsoperationen zwischen der Prozessen, Entwicklung des virtuellen Verbindungsnetzwerkes mit minimal möglichem Zeitaufwand für den Datenaustausch.

Weitere Schritte in Abb.4 werden als der Devirtualisierungsvorgang betrachtet [11].

Anforderungen an die Simulationsmittel

Dynamische Systeme mit konzentrierten und mit verteilten Parametern stellen die folgenden Anforderungen an die Simulationsmittel [1, 3, 5, 7, 11, 13, 14, 15]:

- Berücksichtigung der Tatsache, daß die **formale Beschreibung** komplexer dynamischer Systeme (**KDS-Modelle**) aus einer Topologiedarstellung sowie der großdimensionale Systeme von algebraischen, gewöhnlichen und partiellen Differentialgleichungen besteht;

- **Benutzerfreundlichkeit:** die hochentwickelte Benutzeroberfläche (BOF), rechnerunterstützte Modellerstellung und Modelldebugging für DSKP, DSVP, die dem Benutzer zur Verfügung stehende Simulationssprache mit einer domänenabhängigen Modellspezifikation und leichter Erlernbarkeit;

- **Hochintellektuelle interaktive Unterstützung** des Modellentwicklers (Benutzers) bei der Modellierung und Simulation von KDS:

- Spezifikation der Simulationsobjekte durch die für Experten der Gegenstandgebiete klare Sprachmittel; Minimierung des Aufwandes von Routinearbeiten bei der ursprünglichen Beschreibung und Kodierung der KDS-Topologien;
- Topologische Analyse der Spezifikationen und automatische Generierung der formalen Darstellung von KDS-Strukturen und ihren Parametern;
- Automatische Generierung der KDS-Gleichungssysteme nach Ergebnisse der Topologieanalyse (KDS-Modell) und eine Umwandlung der Gleichungen zur für die

numerische Lösung geeigneten Form (KDS-Simulationsmodell);

- Implementierung der von Modellentwickler ausgewählten numerischen Verfahren und Lösen der Gleichungssysteme (KDS-Simulator);
- Planung und Durchführung von Simulationsexperimenten entsprechend dem KDS-Untersuchungsprogramm;
- Visualisierung und Dokumentierung der Simulationsergebnisse;
- Archivierung der KDS-Modelle für ihre Wiederverwendung

- **Objektorientierte Modellspezifikation** für Struktur, Verhalten und Parameter von DSKP und DSVP, objektimplementierte Modellierungs- und Simulationssoftware;

- **Vorhandensein der Bibliothek von numerischen Verfahren** für die Modellierung und Simulation der komplexen industriellen DSKP und DSVP;

- **Moderne Mittel der Visualisierung und Dokumentation** der Modelle, Simulationsmodelle und Simulationsergebnisse;

- **Vorhandensein der HW/SW-Ressourcen**, die für Simulation der real komplexen DSKP, DSVP genügend sind; Echtzeitfähigkeit zur Kopplung mit den realen DS- und Automatisierungskomponenten (Hardware in-the-Loop);

- **Integrationsfähigkeit** mit den Werkzeuge der rechnergestützten Projektierung von DSKP und DSVP, eine Modellunterstützung aller Etappen der KDS-Projekte;

- **Möglichkeit zur Erstellung** von anwendungsorientierten Simulationsumgebungen;

- **Vorhandensein der Mittel für die Schulung** von Modellentwicklern und Realisierung der Trainingssimulatoren;

- **Eine Systemorganisation** der Modellierungs- und Simulationsressourcen **aufgrund der modernen Informationstechnologien, Integration mit den GRID- und CLOUD-Infrastrukturen.**

Konzeption der verteilten parallelen Simulationsumgebung

Zur Bearbeitung komplexer DSKP-, DSVP-Simulationsaufgaben wird eine verteilte parallele Simulationsumgebung (VPSU) vorgeschlagen, entwickelt und implementiert.

Definition 13: Die VPSU (Abb.5) definieren wir als eine benutzerfreundliche simulationstechnische Systemorganisation der Hardware, der architekturrelevanten Systemsoftware, der speziell zielgerichtet entwickelten Modellierungs- und Simulationssoftware sowie der modellierungs-, simulations- und

systemtechnischbedingten Subsysteme, mit denen alle Etappen der parallelen Modellierung und Simulation von DSKP, DSVP mit der möglichst vollen Berücksichtigung der obigen Anforderungen unterstützt werden.

Die VPSU-Entwicklung wird als zentrales Problem der parallelen Simulationstechnik angesehen und von den beteiligten Institutionen nach der folgenden Konzeption durchgeführt:

- Berücksichtigung der DS-Anforderungen an die Simulationsmittel;
- Nutzung der vorhandenen und künftigen parallelen Ressourcen von verschiedenen Architekturen als verteilten Rechensysteme;
- Entwicklung der funktionsvollständigen VPSU-Modellierungs- und Simulationssoftware entsprechend den formalen Beschreibungen von DSKP, DSVP und DS-Anforderungen;
- Verwendung von denselben Lösensalgorithmen für die diskreten DSKP- und DSVP-Simulationsmodellen;
- Parallele Simulationssoftware mit ähnlichen Eigenschaften wie bei den vorhandenen block- und gleichungsorientierten Simulationssprachen;
- Problemorientierte VPS-Umgebungen werden als Teilprojekte betrachtet, die allgemeine Lösungen nutzen und Ergebnisse für mögliche Verallgemeinerungen liefern;
- VPSU wird als komplexes HW/SW-SYSTEM betrachtet, deren Entwicklung soll auf dem OO-Ansatz mit der UML-Technologie sowie mit der Dekomposition auf bestimmte Menge der funktionellen Subsysteme erfolgen [7]. Solche Entwicklungsorganisation erlaubt nebenläufige Bearbeitung der Teilprojekte (Entwicklungsaufgaben) und eine Erhaltung der Zwischenergebnisse, die zu den arbeitsfähigen Komponenten der Simulationsumgebung führen.

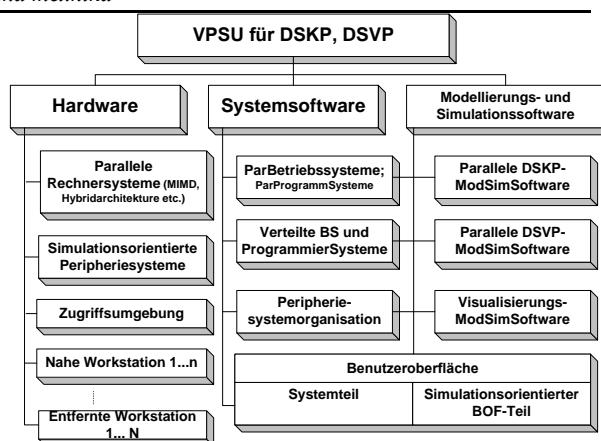


Abbildung 5. Systemorganisation der VPSU

Definition 14: Ein **VPSU-Subsystem** ist eine Teilkomponente von Hardware, Systemsoftware, Modellierungs- und Simulationssoftware, die benutzerfreundlich eine gewisse Gruppe von inhaltlich nahen Funktionen auf entsprechenden DSKP/DSVP-Modellierungs- und Simulationsetappen erfüllt.

Unsere Realisierungserfahrung der VPSU bestätigt die Produktivität und Entwicklungsperspektive vorgeschlagener Konzeption. Die SIMD-Erfahrung [2,4] wird mit den Anwendungsperspektiven von hybriden MIMD-Architekturen [15,16] aktualisiert. Die vorgeschlagene VPSU wird von HLRS und FCWT nach Abb.6 realisiert, implementiert und experimentell untersucht.

Probleme der Entwicklung von VPSU- Modellierungs- und Simulationssoftware

Die Funktionalität der Modellierungs- und Simulationssoftware der Simulationsumgebung wird von in Abbildungen 3, 4 zusammengefassten Etappen der parallelen Modellierung und Simulation von komplexen dynamischen Systemen bedingt. Von der obigen Konzeption der vollfunktionellen VPSU-Entwicklung wird eine Verteilung der gesamten VPSU-Funktionalität zwischen den folgenden Subsystemen vorgesehen:

1. **Dialogsystem** (DiSuS, die Benutzeroberfläche - BOF) – eine Präsentation der VPSU; Schulung von Benutzern; aktives Dialog der Modellentwickler mit alle Ressourcen des verteilten Rechensystems; Integration mit alle anderen Subsysteme; Anpassung an die Simulationsaufgaben der Gegenstandsgebiete; Planung und Führung der Simulationsexperimente.
2. **Topologieanalysesubsystem** (TASuS) – verbale und grafische Darstellung sowie geeignete Kodierung von ursprünglichen VTF-, SAS-, DNO-Topologien; Dekomposition und Ap-

proximation, Erstellung der sekundären Topologien; Formierung der topologierelevanten Vektoren und Matrizen; Umwandlung der ursprünglichen Kodierungstabellen in die Zwischenformen; Darstellung der topologischen Informationseinheiten in der für die Gleichungsgenerierung geeigneten Form; eine Ausgabe der Ergebnissen von Topologieanalyse.

3. **Subsystem der Gleichungsgenerierung (SuGG)** – Kommunikation mit dem TASuS; Darstellung der ursprünglichen Gleichungen (Modelle) in Vektormatrixform; formale Vektor-Matrix-Operationen der Umwandlung „DSKP-Modell – Simulationsmodell“; Darstellung der approximierten DSVP-Gleichungen (1D-, 2D-, 3D-ortsdiskretisierten Modelle) in der Multi-vektor-Matrix-Form; die formale Umwandlung „DSVP-Modell – Simulationsmodell“; die Generierung von diskreten DSKP-, DSVP-Simulationsmodelle für gegebenes numerisches Lösungsverfahren; die Visualisierung der Modelle und Simulationsmodelle.

4. **Subsystem der virtuellen parallelen Simulationsmodelle (VPSMSu)** – interaktive Darstellung der Hierarchie von virtuellen parallelen Simulationsmodelle abhängig von möglichen Parallelisierungsansätze; TASuS-gestützte Erstellung der Topologien entsprechend den VPSM-Ebenen; SuGG-unterstützte Formierung der VPSM-Ebene-Gleichungssysteme; Apriori-Analyse der diskreten VPSM aller Ebenen; interaktive Bereitstellung von implementierungsfähigen diskreten VPSM.

5. **Subsystem von parallelen Gleichungslösern (SuGL)** – Kommunikation mit dem TASuS und VPSMSu, die Eingabe der implementierungsfähigen diskreten VPSM; die Lösung von algebraischen, gewöhnlichen und partiellen Differentialgleichungssysteme mit Hilfe der in den DSKP-, DSVP-Bibliotheken zusammengefassten parallelen numerischen Verfahren; die Analyse der Konvergenz, Stabilität und Genauigkeit der Lösungen, Optimierung der veränderlichen Parametern; Abschätzung und Optimierung der Effizienzmerkmalen von parallelen Lösungen in Vergleich mit den sequentiellen Verfahren; Formierung der Lösungsergebnisse für die anschauliche Präsentation.

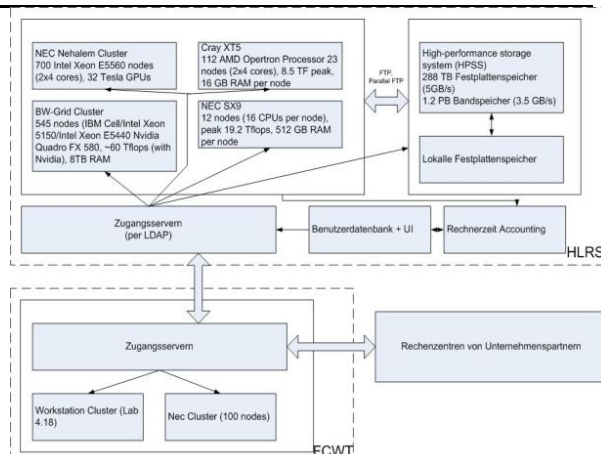


Abbildung 6. Aktueller Stand der von der FCWT und vom HLRS entwickelten VPSU-Hardwarestruktur

6. **Datenaustauschsystem (DASu)** – die vollfunktionelle Liste der angemeldeten VPSU-Teilnehmern und vorhandenen VPSU-Ressourcen, die ein Datenaustausch nach gemeinsame Initiative durchführen sollen; eine Hierarchiedarstellung von Datenströme für ausgewählten ZRS; Testsystem für die Ermittlung der realen Parametern von Austauschoperationen im ZRS-Verbindungsnetzwerk; Optimierung der parallelen Programme bezüglich der Datenaustauschoperationen; Zusammenstellung der in der Apriori-Analyse und im ZRS erhaltene Wirkung der DA-Operationen auf die Werte der Effizienz von Parallelisierungsansätzen.
7. **Lastbalancierungssystem (LaBSu)** – Abschätzung der Lasthöhe von virtuellen Prozessen in VPSM-Ebenen, statische Lastbalancierung der VPSM aller Ebenen; Spezifikation der Aufträge auf VPSU-Ressourcen von implementierten VPSM; eine Ermittlung der Lastverteilung zwischen Prozessen und zwischen ZRS-Prozessoren; Vergleichsanalyse der Parallelisierungsansätze nach Kriterium der gleichmäßigen Lastverteilung.
8. **Visualisierungssystem (ViSuS)** – Visualisierung der Ergebnisse von Topologieanalyse und Gleichungsgenerierung; in Integration mit BOF und Subsystem Gleichungslösern eine Vorbereitung und Strukturierung der Simulationsergebnissen zu den Formen, die für grafischen 1D-, 2D- und 3D-Visualisierung geeignet sind; interaktive Erstellung von Grafiken während und nach der Durchführung von Simulationsexperimenten.
9. **Datenbanksystem (DaBSu)** – in der Integration mit allen anderen Subsystemen: Daten über VPSU-Hardware und Systemsoftware; Benutzerdaten; ursprüngliche und umgewandelte Daten der modellierten VTF-, SAS- und DNO-KDS; Archi-

vierung der lauffähigen parallelen und sequentiellen Programme; Daten der Testaufgaben, Pläne der Simulationsexperimente, Archivierung der Simulationsergebnissen.

10. **Subsystem der IT-Unterstützung (SuIT)** – Betriebsorganisation von verteilten Rechen-, Kommunikations- und Simulationsressourcen durch die Technologien der modernen Netzwerke; entfernter WEB-basierter Ansatz zur parallelen Modellierung und Simulation; Funktionierung der Ketten „WEB-Client (Modellentwickler) – WEB-Server – PARSIMULATOR“ und „WEB-Client (Modellentwickler) – WEB-Server – DATABASE-Server“; Integration mit anderen Subsystemen und mit den GRID- und CLOUD-Infrastrukturen [13,14].

Die vollfunktionelle DSVP-Simulationssoftware soll die in Abb.7 gezeigten Module beinhalten.

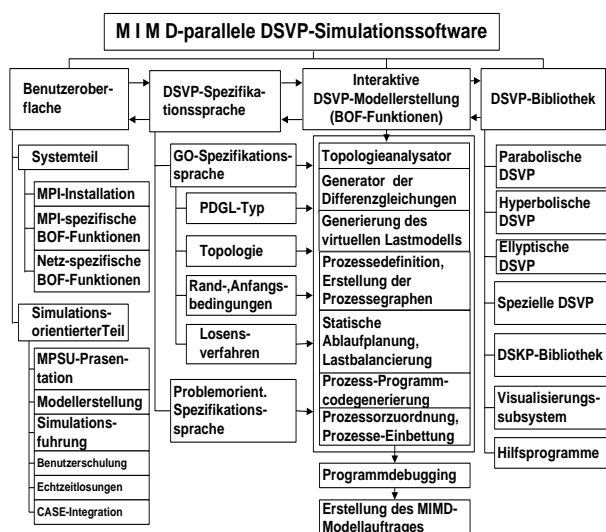


Abbildung 7. Struktur der parallelen Modellierungs- und Simulationssoftware für dynamische Systeme mit verteilten Parametern

Sie entsprechen den Etappen der parallelen DSVP-Modellierung und Simulation (Abb.3, 4) und der obigen Anforderungen an die Modellierungs- und Simulationsmitteln. DSVP-Bibliothek stellt dem Benutzer die parallelen numerischen Lösungsverfahren der partiellen Differentialgleichungen der genannten Typen zur Verfügung. Die DSKP-Simulationssoftware besitzt eine ähnliche Struktur. Die parallele numerische Verfahren fürs Lösen der gewöhnlichen Differentialgleichungen werden in DSKP-Bibliothek zusammengefasst und mit DSVP-Bibliothek integriert, weil die Dekomposition und Ortsdiskretisation von DSVP zu den großdimensiona-

len Systemen der gewöhnlichen und algebraischen Gleichungen führt.

Die in obigen Bibliotheken eingebetteten parallelen Gleichungslösern werden nach folgenden Ansätzen entwickelt:

1. Eine Parallelisierung der numerischen Verfahren, die in den benutzerfreundlichen konventionellen Simulationssprachen erfolgreich verwendet werden (Euler, Adams-Bashforth, Runge-Kutta u.a. Verfahren). Diskrete Form der minimalkörnigen Prozessen und virtuellen parallelen Simulationsmodelle ist durch die Berechnungsformeln dieser Verfahren definiert.
2. Entwicklung der neuen parallelen Lösungsverfahren.

Für die gewöhnlichen Differentialgleichungssysteme werden die blockartigen Verfahren [17,18] entwickelt, welche für einen Block von k Punkten berechnen die neue k Funktionswerte gleichzeitig. Diese Verfahrenseigenschaft passt gut zu parallelen Rechnerarchitekturen und erlaubt die Berechnungen von Koeffizienten der Differenzformeln vor der Integration, schon während der Entwicklung des Verfahrens und erhöht die Effizienz des Rechnens. Betrachten wir das Lösen des Cauchy-Problems

$$x' = f(t, x), \quad x(t_0) = x_0. \quad (1)$$

mit Hilfe des blockartigen k -Punktverfahrens (Abb. 8). Zerlegen wir dabei die Knotenmenge M des homogenen Gitters $\{t_m\}$, $m = 1, 2, \dots, M$ mit dem Schritt τ auf Blöcke, die je k Punkte beinhalten, $kN > M$. Numerieren wir in jedem Block die Punkte $i = 0, 1, \dots, k$ und bezeichnen als $t_{n,i}$ den Punkt i des n -ten Blocks. Die Punkte $t_{n,0}$ und $t_{n,k}$ sind entsprechend der Anfang und das Ende des Blocks n . Daraus folgt, dass $t_{n,k} = t_n + t_{n,0}$, dabei wird der Anfangspunkt dem Block nicht gehören.

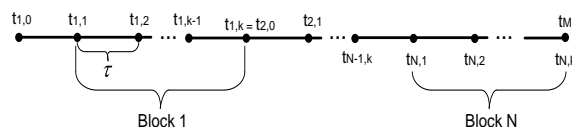


Abbildung 8. Schema der Zerlegung auf die Blöcke

Wir werden zwischen den blockartigen Einzschritts- und Mehrschrittsverfahren unterscheiden (Abb. 9,10).

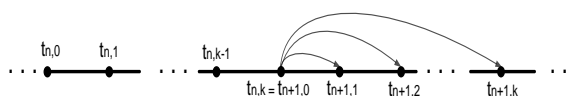


Abbildung 9. Berechnungsschema beim blockartigen Einzschrittsverfahren

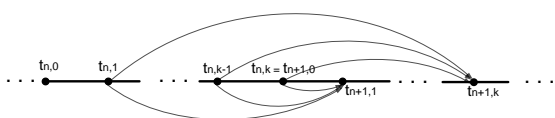


Abbildung 10. Berechnungsschema beim blockartigen Mehrschrittsverfahren

Das m -schrittliche k -punktliche blockartige Verfahren hat folgende allgemeine Form der Berechnungsgleichungen[17]:

$$u_{n,i} = u_{n,0} + i\tau \left[\sum_{j=1}^m b_{i,j} F_{n,j-m} + \sum_{j=1}^k a_{i,j} F_{n,j} \right], \quad (2)$$

wo $i = \overline{1, k}$ $n = \overline{1, N}$ $F_{n,j} = f(t_n + j\tau, u_{n,j})$.

Es wurden die ausführlichen theoretischen Untersuchungen der blockartigen Ein- und Mehrschrittsverfahren bezüglich der Konvergenz- und Stabilitätsanalyse, der Fehlerabschätzungen durchgeführt [17,18]. Aufgrund der impliziten Einzelschrittsverfahren wurden die parallele blockartige eingebetteten und extrapolierende Verfahren für das beschleunigte Lösen der steifen Gleichungen vorgeschlagen und begründet. Die hergeleiteten analytischen Ausdrücken für die Abschätzung der Beschleunigung, Effizienz und Skalierbarkeit der parallelen blockartigen Ein- und Mehrschrittsverfahren wurden auf Testaufgaben erprobt. Diese Ergebnisse stellen zuverlässiger Grund für die weiteren Entwicklung und Realisierung paralleler Gleichungslösern und Simulatoren dar [20].

Problemorientierten parallelen Simulationsumgebungen

Entwicklung der problemorientierten VPSU bleibt als aktuelle Herausforderung in verschiedenen Gegenstandsgebiete [1, 5, 9, 10, 12, 19, 20]. Wissenschaftliche und praktische Bedeutung haben folgende Aufgabenlösungen: die Spezifikationssprachen der DSKP, DSVP auf technologischer Ebene; die formale Beschreibung von Topologien und physikalischen Prozessen, Entwicklung der Algorithmen der Topologieanalyse und der Gleichungsgenerierung; die Ansätze zur Parallelisierung, Entwicklung der virtuellen parallelen Simulationsmodelle aufgrund der blockartigen Verfahren (BAV) und ihre theoretische Analyse; Devirtualisierung und Prozessezuordnung auf ZRS, Anpassung der numerischen Verfahren und Aufbau von parallelen Gleichungslösern; Entwicklung der effektiven Subsysteme; Effizienzanalyse und Optimierung der parallelen Simulatoren; Integration mit der gegenstandsgebietspezifischen CASE-Werkzeuge, modellgestützte Projektierung und Prozessführung; benutzerfreundliche Systemorganisation der Anwendung von parallelen Ressourcen.

Simulationsumgebung für dynamische Netzobjekte (DNO-Topologie)

Die dynamischen technischen und verfahrenstechnischen Netze sind in mehreren Gegenstandsgebiete die Objekte der Untersuchung, Projektierung, Automatisierung, Überwachung, Qualitätssicherung, optimalen Prozessführung, Sicherheitsanalyse und -vorhersage, Vermeidung der sicherheitskritischen Betriebszustände und der Havarieliquidierung. Die Netzobjekte gehören zu den komplizierten (oft sicherheitskritischen) dynamischen Systemen. Nichtlinearität der prozessbeschreibenden Funktionen, räumliche Verteilung von Prozessparametern, große, im Laufe des Objektbetriebs sich entwickelnde Dimensionen der Netze (Kantenmenge $m > 100$, Knotenmenge $n > 50$), ebenso mehrere aktive Elemente mit nichtlinearen stromabhängigen Charakteristiken, wesentliches mehrfaches und hierarchisches Zusammenwirken der regelbaren Prozessparametern, sowie gleichzeitige Einflüsse von deterministischen und stochastischen Störungen sind die Hauptmerkmale dieser Komplexität. Nur wenige, sehr vereinfachten Aufgabe der dynamischen Netzobjekte kann man jedoch analytisch lösen. Deshalb haben die Methoden und Mittel der Modellierung und Simulation dieser Objektklasse sowohl bei der Projektierung als auch während des Betriebes eine zunehmende theoretische und praktische Bedeutung. Als DNOVP-Beispiel betrachten wir ein Grubenbewerternetz. Topologisch wird DNOVP als Graph $G(m, n)$ dargestellt und durch Tabelle 1 mit m Zeilen und $s+5$ Spalten kodiert. Hier sind: QJ – ein Luftstrom in der J-Kante; AKI und EKK – Anfangs- und Endknoten der J-Kante; $(I, K) \in (1, 2, \dots, n)$, $J \in (1, 2, \dots, m)$; $PAR(PJ_1, PJ_2, \dots, PJ_s)$ – Menge von s Parametern PJ der Kante; AEJ – aktives Element in der J-Kante; VECOMJ – Kommentar zur technologische Funktion der J-Kante.

Tabelle 1. Kodierung des Graphen

AKI	EKK	QJ	PAR(PJ ₁ , PJ ₂ , ..., PJ _s)	AEJ	VECOMJ
-----	-----	----	--	-----	--------

Modell der dynamischen Prozesse in der j -Kante ohne Luftverlusten durch die Wände wird von den Gleichungen

$$\begin{cases} \frac{\partial P_j}{\partial \xi} = r_j Q_j + \frac{\rho Q_j}{F_j} \frac{\partial Q_j}{\partial \xi} + r_j (\xi \tau) \} \\ \frac{\partial P_j}{\partial \xi} = \frac{\rho \partial Q_j}{F_j \partial \xi} \end{cases} \quad (3)$$

beschrieben. Hier sind: P_j, Q_j – Druck und Luftstrom der Koordinate ξ entlang, die von AKI- bis zum EKK-Knoten errechnet wird ; r_j – spezifischer aerodynamischer Widerstand; F_j – die Querschnittsfläche

der Kante (Luftwegstrecke); ρ – Luftdichte; a – die Schallgeschwindigkeit im Luft; $r_j(\zeta, t)$ – regelbarer Widerstand; ζ_j – die Ortskoordinate des regelbaren Widerstands (z. B., ein Schieber).

Die Randbedingungen für (3) sind die Druckfunktionen P_{AKI}, P_{EKK} in den Knoten der j -Kante. Es sind drei Kanten- und Knotenarten nach Randbedingungen in DNOVP zu unterscheiden:

- die Kanten, die den inneren n_1 DNO-Knoten inzident sind; hier werden die Druckwerte während des Lösens des DNOVP-Gleichungssystems entsprechend den dynamischen Knotenbedingungen

$$\frac{\partial P_{wi}}{\partial t} = \frac{Q_{wi}}{F_{wi}} \Delta \xi_{jk} \quad (4)$$

berechnet; hier sind: P_{wi} – Druck im WI -Knoten; Q_{wi} – Gesamtluftstrom durch WI -Knoten; F_{wi} – Querschnittsfläche des Knotenraums;

- die Kanten, die den n_2 Knoten der Ventilatorenanschlüsse inzident sind; hier wird Druck als die Ventilatorcharakteristik vorgegeben

$$P_{wi} = P_{AEJ}(QJ); \quad (5)$$

- die Kanten, die den n_3 Knoten der Atmosphäreanschlüsse inzident sind:

$$P_{wi} = P_{ATM} = const. \quad (6)$$

DNOVP hat insgesamt

$$n = n_1 + n_2 + n_3 \quad (7)$$

Knoten und entsprechend die n Randbedingungen. Die Anfangsbedingungen sind

$$P_j(\zeta, 0), Q_j(\zeta, 0) \quad (j = 1, 2, \dots, m). \quad (8)$$

Problemstellung: für DNOVP, dessen Graph wird mit der topologischen Tabelle 1 kodiert und jede Kante wird mit den Gleichungssysteme (3) und Randbedingungen (4), (5), (6) bei den Anfangsbedingungen (8) beschrieben, sollen die parallelen algorithmischen, hardware- und softwaretechnischen Simulationsmitteln entwickelt und implementiert werden, die adäquat die dynamischen Prozesse $P_j(\zeta, t), Q_j(\zeta, t)$ ($j = 1, 2, \dots, m$) mit der Berücksichtigung von definierten Arbeitsbedingungen, Störungen und Regelungen von Luftströmen widerspiegeln.

Durch die Approximation der Gleichungen (3) nach **Linienverfahren** mit der Ortschrittweite $\Delta \zeta$ erhalten wir für k -Element der j -Kante das Gleichungssystem:



$$\frac{dP_{wi}}{dt} = \frac{Q_{jkw} - Q_{jwi}}{F_{wi}} \Delta \xi_{jk} \quad (9)$$

Die $\alpha_j, \beta_j, \beta_{ij}, g_j$ sind die von aerodynamischen j -Kanteparametern abhängigen Koeffizienten.

Die inneren Randbedingungen von Type (4) werden so dargestellt:

$$-\frac{dP_{wi}}{dt} = \frac{\rho a^2}{F_{wi}} \frac{Q_{jk} - \sum_{jwi} (Q_{jwi1} - Q_{jwiMj})}{\Delta \xi_{jk}} \quad (10)$$

Hier sind: $P_{wi} = P_{jMj+1}$ – Druck im Endknoten des letzten Elementes $Q_{jk} = Q_{jMj}$ des J -Kantestroms, der in wi -Knote fließt; $jwiMj$ – die Kantenummern aus Menge $j=1, 2, \dots, m$, die dem Knoten wi inzident sind; dabei $jwi1$ ist erstes Element der j -Kante mit den wi als Anfangsknoten (Ausfluß), während $jwiMj$ letztes Element mit den wi als Endknoten (Zufluß) ist. Jede Kante wird bei der Approximation nach M_j Elementen Q_{j1}, \dots, Q_{jMj} zerteilt. Dabei wird die Nummerierung von Druckwerten als $P_{j1}, P_{j2}, \dots, P_{jMj+1}$ erfolgt. Es ist wichtig zu erwähnen, dass j -Kante eine Anfangsknote wi mit dem Druck $P_{wi} = P_{j1}$ und die Endknote $wi+b$ ($b = const$) mit dem Druck $P_{wi+b} = P_{jMj+1}$ hat. Für dynamisches Netzobjekt mit verteilten Parametern wird jede Kante nach obige Approximation durch zwei Vektoren Q_j, P_j ($j = 1 \dots m$) präsentiert:

$$Q_j = (Q_{j1}, Q_{j2}, \dots, Q_{jMj})^T \quad (11)$$

ist Luftstrom in j -Kante,

$$P_j = (P_{j1}, P_{j2}, \dots, P_{jMj+1})^T \quad (12)$$

ist Druck in j -Kante. Dabei wird M_j – die Menge der Elementen in den Kanten – abhängig von der Kantentlängen l_j bei der gleichen Ortschrittweite $\Delta \xi$ für ganzes DNOVP als $M_j = l_j / \Delta \xi$ berechnet.

Bei der Entwicklung des ortsdiskretisierten Simulationsmodells des ganzen Netzobjektes sollen wir m Gleichungssysteme von Type (9) für alle Kanten, d. h. $j=1, 2, \dots, m$, darstellen:



$$\frac{dP_{wi}}{dt} = \frac{Q_{jkw} - Q_{jwi}}{F_{wi}} \Delta \xi_{jk} \quad (13)$$



$$\frac{dP_{wi}}{dt} = \frac{Q_{jkw} - Q_{jwi}}{F_{wi}} \Delta \xi_{jk} \quad (14)$$

Entsprechend der Gleichung (10) formulieren wir n_1 Randbedingungen für inneren Knoten des Netzobjektes ($wi=1, 2, \dots, n_1$):



$$\frac{dP_{wi}}{dt} = \frac{Q_{jkw} - Q_{jwi}}{F_{wi}} \Delta \xi_{jk} \quad (14)$$



Die Ventilatorencharakteristiken bilden nach (5) n_2 weitere Randbedingungen

$$P_{wi} = P^1_{AEJ}(QJ)$$

$$\dots\dots\dots (15)$$

$$P_{wi} = P^{n_2}_{AEJ}(QJ).$$

Dabei ist $wi = n_1 + 1, n_1 + 2, \dots, n_1 + n_2$, wenn die Nummerierung der Knoten fortlaufend von inneren Knoten durchgeführt wird. Aktive Elemente (Ventilatoren) sind in (15) von 1 bis n_2 zusätzlich nummeriert.

Für n_3 Knoten der Atmosphäreanschlüsse gelten die Randbedingungen nach (6):

$$P_{wi} = P_{(n_1+n_2)+1} = P^1_{ATM} = const,$$

$$\dots\dots\dots (16)$$

$$P_{wi} = P_n = P^{n_3}_{ATM} = const.$$

Die Gleichungen (13) bilden zusammen mit den Randbedingungen (14), (15), (16) das ortsdiskretisierte DNOVP-Simulationsmodell. Für die industrienahen DNOVP ($m \geq 1000, n \geq 300, M_j \geq 50$) ist aktuell die rechnergestützte Erstellung der DNOVP-Simulationsmodelle mit Hilfe des Topologieanalytators und des Gleichungsgenerators zu realisieren.

In Abb. 11 werden die Dekomposition des Grubenbewerungsnetzes und Diskretisierung der Luftwege (Kanten des Netzgraphen) erläutert. Die Parallelisierungsansätze geben vier möglichen Parallelitätsebenen des virtuellen parallelen Simulationsmodells (Abb.12). Es wurde paralleler DNOVP-Simulator (Abb.13) entwickelt, implementiert und im Simulations- und Servicezentrum für die Kohleindustrie [9,10] experimentell untersucht.

Abbildung 11. Dekomposition und Diskretisierung des dynamischen Netzobjektes

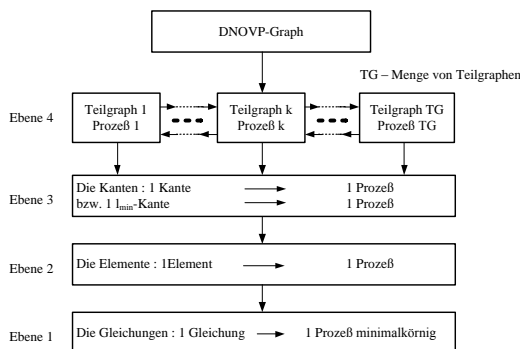


Abbildung 12. Die virtuellen Parallelitätsebenen des DNOVP-Simulationsmodells

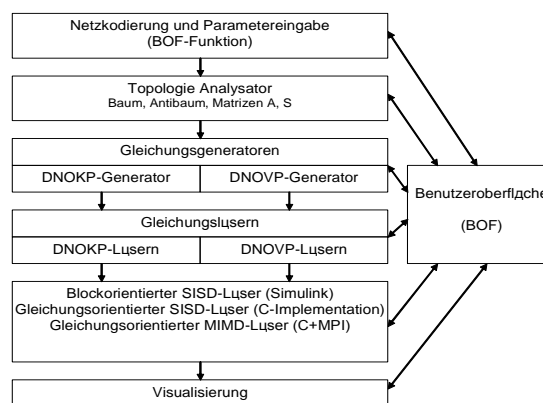


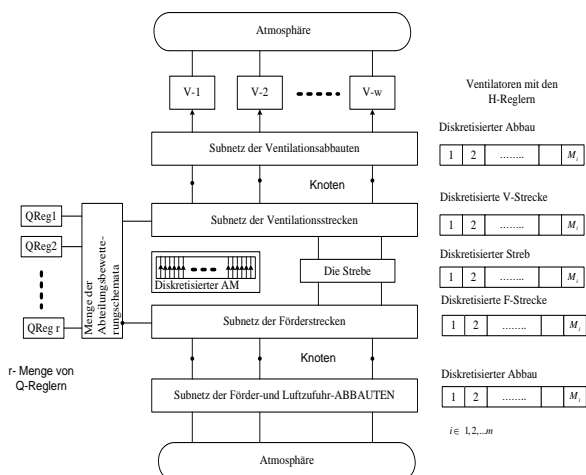
Abbildung 13. Struktur des DNO-Simulators

Verfahrenstechnisch orientierte parallele Simulationsumgebung

Basierend auf der Entwicklungserfahrung von universellen und problemorientierten parallelen Simulationsumgebungen für dynamische Systeme mit konzentrierten und verteilten Parametern, der Entwicklung und Realisierung der Modellierungssystematik von verfahrenstechnischen Prozesse und Anlagen (VTPA) in dem Simulationssystem DIVA und im Werkzeug PROMOT [1, 5], einer Analyse der Simulationsstrategien und modularen Simulationsansätze in der Verfahrenstechnik [1, 5] wird eine verteilte verfahrenstechnisch orientierte parallele Simulationsumgebung (VTOPSU) vorgeschlagen.

VTOPSU entspricht der Definition 13 und wichtige Eigenschaften der Verfahrenstechnik als des Gegenstandsgebietes berücksichtigt. Abb. 14 zeigt die Strukturdarstellung der VTOPSU. Es ist eine Versuchsversion der Umgebung realisiert und am Beispiel der Modellierung und Simulation der Essigsäureproduktion erprobt.

Die vorgesehene VTOPSU-Funktionalität umfasst das rechnergestützte dialogbasierte Lösen folgenden Aufgabenblöcke bei der parallelen Modellie-



rung und Simulation von verfahrenstechnischen Prozessen und Anlagen:

- Darstellung der VTP/VTA-Topologie, eine Generierung der topologie-bedingten Gleichungssysteme in der von numerischen Lösungsverfahren geforderte Form;
- Erstellung der virtuellen parallelen VTP/VTA-Simulationsmodelle mit der Be-

rücksichtigung der alternativen Parallelisierungsansätze;

- Aproiri-Analyse und Devirtualisierung der Simulationsmodelle und ihre Zuordnung auf paralleles Zielrechnersystem;

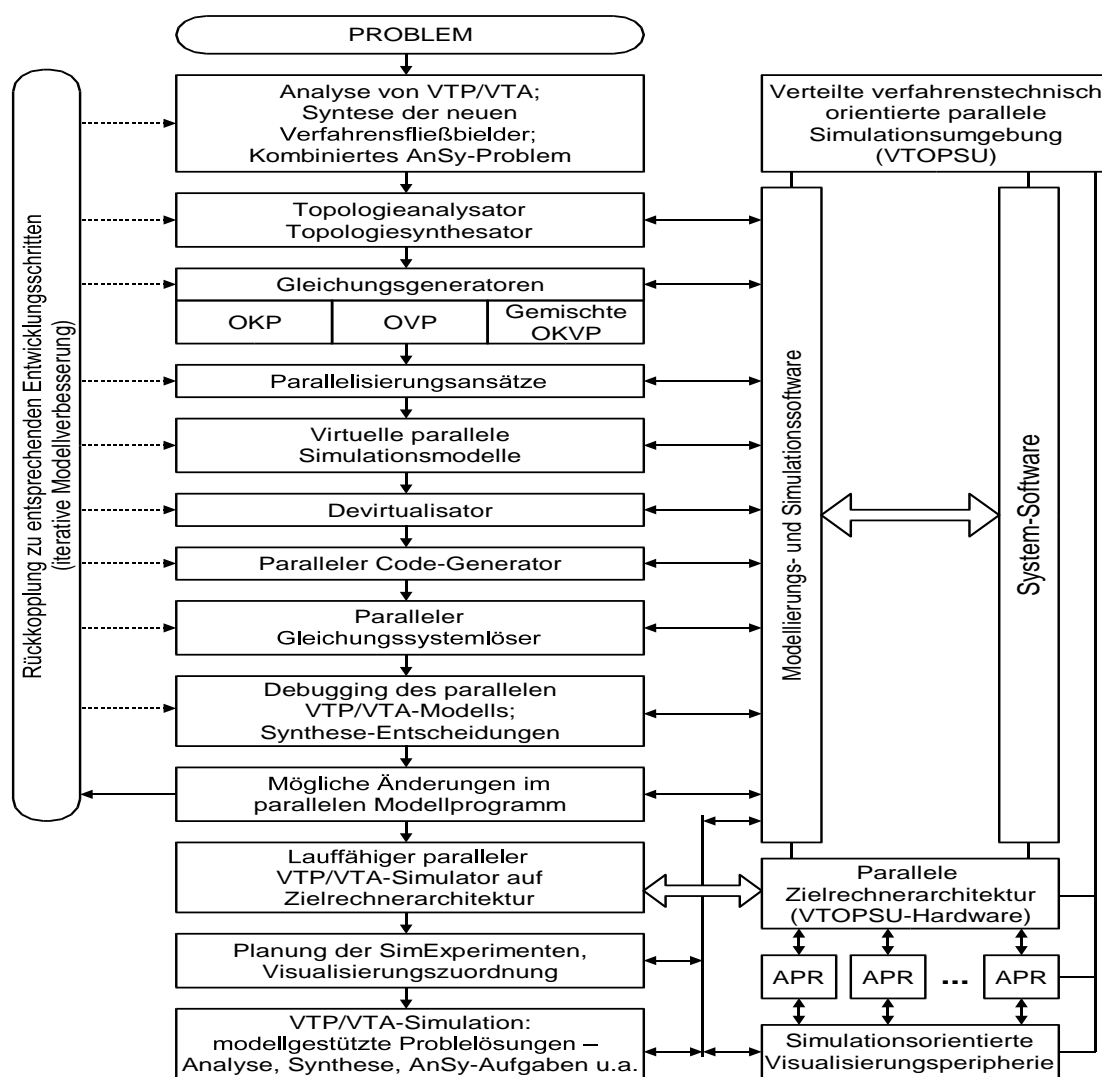


Abbildung 14. Strukturdarstellung der VTOPSU

- Generierung des parallelen Programmcodes für dem ZRS zugeordnetes Simulationsmodell;
- Initialisierung, Parameterzuweisung und Debugging der parallelen VTP/VTA-Simulationsmodelle, ihre iterative Qualitätsverbesserung;
- Effektive parallele Lösung der für die VTP/VTA-Simulationsmodelle relevanten Gleichungssysteme mit der Visualisierung

der erhaltenen dynamischen Verhalten von Prozessen und Anlagen;

- Aufbau der parallelen VTP/VTA-Modellbibliothek, die Wiederverwendung der parallelen Simulationsmodelle, Realisierung der verfahrenstechnischen Modellierungssystematik in den örtlich verteilten Rechnernetzen.

- **Problem der Entwicklung von parallelen Simulationssprachen**

Die Benutzerfreundlichkeit der parallelen Rechnersysteme ist in der ersten Reihe mit den vorhandenen Programmiermodelle und Mitteln für Realisierung der parallelen Lösungsalgorithmen verbunden. Die Systeme der 90-zigen Jahre verfügten über parallele SIMD- und MIMD-Programmiersprachen, die aufgrund von Fortran, C, C++, Modula-2 u. a. Sprachen gebaut wurden [2]. Intensive Entwicklung der MIMD-Systeme und objektorientierten Ansätze hat die Standardisierung der Technologien der parallelen und verteilten Programmierung stimuliert. So haben ANSI, ISO C++-Standard mit den MPI-, PVM- und Pthreads-Bibliotheken eingeführt. Trotz gewisser Fortschritt soll Gegenstandsexperte (Modellentwickler) heute wie früher mit der Programmiersprache die parallelen Modelle realisieren. Diese Tatsache beschränkt wesentlich eine Nutzung der parallelen Ressourcen von Fachleuten, die mit den block- (BO), gleichungs- (GO) und objektorientierten (OO) Simulationssprachen [1, 3, 8] dynamische Systeme modellieren und simulieren. Um VPSU zu den benutzerfreundlichen Simulationen einzunähern, sollen wir die parallele Modellierungs- und Simulationssoftware in Richtung **parallelen Simulationssprachen weiter zu entwickeln**. Diese Sprachen sollen eindeutige Verbindung zwischen den Gegenstandsgebiete der verfahrenstechnischen Ebene und der formalen Beschreibung von dynamischen Systemen (Abb.1) mit Hilfe der Modellspezifikation benutzerfreundlich widerspiegeln und in den lauffähigen parallelen Simulatoren umwandeln (Abb.2 ,3). Analyse zeigt, dass die VTF-, SAS- und DNO-Topologien allgemein in Graphendarstellung umgewandelt werden können. Die entwickelten und experimentell untersuchten Paare „Topologieanalysatoren - Gleichungsgeneratoren“ geben die Simulationsmodelle, die direkte Anwendung der BO- und GO-Prinzipien der Gleichungslösung erlauben, ohne die diskreten Simulationsmodelle zu generieren. Dabei sind die folgenden Ideen der BO- und GO-Ansätze zu beachten.

BO-Lösensprinzip entspricht der MIMD-Parallelität und kann als eine virtuelle Zuordnung „Funktionsblock – Prozess“ betrachtet werden [21]. Dem minimalkörnigen Prozess soll die Funktionsblockkette zugeordnet werden, welche eine Differentialgleichung der ersten Ordnung löst. Diese Ketten werden vom *Generator des BO-Lösers* nach entwickelten Algorithmen erstellt. So entsteht die von dem BO-Lösensprinzip bedingte virtuelle Blockschaltung, die das virtuelle parallele Simulationsmodell der

ersten Parallelitätsebene (PE-1) mit minimal körnigen Prozessen (MKP) „realisiert“. Das virtuelle Verbindungsnetzwerk (VNW) hat zwei Ebene: innere MKP-Verbindungen sowie äußere, die alle MKP entsprechend der Approximationschema verbinden. Der Devirtualisierungsvorgang wird vom vorgeschlagenen in der VPSU-Simulationssoftware (Abb.3) eingebauten BO-Devirtualisator durchgeführt. Als Randbedingungen hat er PE-1 und ZRS. Die optimale Zuordnung dem ZRS wird mit der Berücksichtigung der höheren Parallelitätsebenen etwa PE-2 – PE-4 und der Abbildung ihren virtuellen VNW-2 – VNW-4 in dem ZRS-Verbindungsnetzwerk realisiert. Diese Zuordnung benutzt Code-Generator, der ein lauffähiges MPI-Programm produziert.

GO-Lösensprinzip basiert sich auf der Darstellung der Simulationsmodelle in Form von Funktionsoperatoren, die als feinkörnige Prozesse betrachtet werden können. Es gibt auch die komplexen Operatoren, die für das Lösen der Differentialgleichungen der 1.Ordnung zuständig sind und dem oben eingeführten minimalkörnigen MIMD-Prozess entsprechen. GO-Ansatz braucht deshalb eine zusätzliche Funktion in den obigen Gleichungsgeneratoren (Abb.3): „Umwandlung des Simulationsmodells in GO-Form“. Als Ergebnis wird virtuelles paralleles GO-Simulationsmodell PE-1 sein. Virtuelles Verbindungsnetzwerk ist mit obigen äußeren BO-Variant ähnlich und realisiert die Verbindungen zwischen den GO-Operatoren-MPK entsprechend der DSVP-Approximation. Wie im BO-Ansatz sollen hier der GO-Devirtualisator und der Code-Generator entwickelt und eingebaut werden.

Objektorientierter Ansatz ist aufgrund der vorhandenen OO-Simulationssprachen [8] mit der Berücksichtigung von obigen beiden Ansätzen realisierbar.

Zusammenfassung und Ausblick

Modernste Simulationstechnik wird von uns in zwei Hauptaspekte betrachtet:

- als interdisziplinäres selbständiges Forschungsgebiet mit den fundamentalen theoretischen Grundlagen und Methoden der Realisierung von universellen Simulationswerkzeugen;
- als angewandte Wissenschaft, die in allen Gegenstandsgebiete für den Aufbau der mathematischen Modelle, Simulationsmodelle und Simulatoren, für die simulationsgestützte Entwicklung der automatisierten verfahrenstechnischen Prozesse, Anlagen und Systeme sowie für die Schulung des Betriebspersonals hilfreich und oft entscheidend ist.

Die steigenden Anforderungen seitens Gegenstandsgebiete an die Methoden und Mittel der Modellierung und Simulation betreffen beide obige Aspekte der Simulationstechnik, stimulieren die Anwendung von parallelen Höchstleistungsrechnern der vorhandenen und künftigen MIMD-Architekturen und rufen die neue zu lösenden theoretischen und praktischen simulationstechnischen Probleme hervor, die wir als Probleme der **parallelen Simulationstechnik (ParSimTech)** definieren:

Simulationstechnisch effiziente Nutzung der parallelen Hardware/Software-Ressourcen: die vorgeschlagene Konzepte der allgemeinen und problemorientierten parallelen Simulationsumgebungen sowie ihre Implementierungen bedeuten eine neue Systemorganisation der parallelen Simulationsmittel.

Erhöhung der Benutzerfreundlichkeit paralleler Simulationswerkzeuge: die vorgeschlagene Topologieanalysatoren und Gleichungsgeneratoren für dynamischen Netzobjekte können in verschiedenen Anwendungsgebieten eingesetzt werden.

Rechnergestützte Erstellung von Modellen, parallelen Simulationsmodellen und Simulatoren für DSKP und DSVP: mit Hilfe der vorgestellten virtuellen Simulationsmodelle und Devirtualisierungsvorgänge erfolgt eine Systematisierung der einzelnen Entwicklungsschritte.

Entwicklung von parallelen Simulationsalgorithmen für DSKP und DSVP: die entwickelten Gleichungslösern aufgrund der konventionellen und parallelen numerischen Verfahren sind wichtige Bestandteile der parallelen Simulationsumgebungen.

Schnelle Implementierung und Einsatz der problemorientierten parallelen Simulationsumgebungen: die untersuchte PSU-Dekomposition auf funktionelle Subsysteme erlaubt eine parallele system- und programmtechnische PSU-Entwicklung.

Entwicklung der parallelen Simulationsprachen als wichtiger Faktor für die Erweiterung der Anwendung von parallelen Rechnern und der modellgestützten Projektierung: Die betrachteten Ansätze erlauben den kontinuierlichen Übergang von vorhandenen standardisierten Sprachen und Bibliotheken der Programmierung zu den höheren Sprachen der parallelen Modellierung und Simulation.

Integration der Methoden und Werkzeugen der parallelen Simulationstechnik mit der Theorie und Praxis von verteilten Rechensystemen und modernen IT-Lösungen: Implementierungserfahrungen mit der örtlich verteilten parallelen Simulationsumgebung und ein effektiver Fernzugriff auf verschiedene Parallelrechner bilden eine Basis für experimentelle Untersuchungen in Richtung der Integration von verteilten parallelen Simulationsumgebungen mit den GRID- und CLOUD-Infrastrukturen [22].

Die geplante weitere Entwicklung der parallelen Simulationstechnik konzentriert sich auf parallelen blockartigen Gleichungslösern, parallelen Simulationsalgorithmen und Simulationssprachen, objektorientierten architekturabhängigen hierarchischen Parallelisierungsansätze, problemorientierten Umgebungen sowie auf industrielle und universitäre Anwendungen.

Literatur

1. Zeitz, M.: Simulationstechnik. Chemie-Ingenieur-Technik, 59, 1987, 464-469.
2. Feldmann, L., Resch, M., Svjatnyj, V., Zeitz, M.: Softwarearchitektur für parallele Simulationsumgebungen. Plenarvortrag, ASIM'2014 in Berlin, Tagungsband 1, ARGESIM Report 43, Wien 2014, S. 3-7.
3. Kramer, U., Neculau, M.: Simulationstechnik. Carl Hanser Verlag, München- Wien 1998.
4. Feldmann L.P. Svjatnyj V.A., Bräunl T., Reuter A., Zeitz M.: Implementierung und Effizienzanalyse der parallelen Simulationsalgorithmen für dynamische Systeme mit verteilten Parametern (Plenarvortrag). Tagungsband 11. Symposium ASIM'97, Vieweg 1997, 38-47.
5. Gilles, E.-D.: Modellierungssystematik für verfahrenstechnische Prozesse. In: Abschlussbericht "Methoden zur Modellierung und Berechnung der Dynamik verfahrenstechnischer Prozesse" der DFG-Forschergruppe Universität Stuttgart, 1996, 7-22.
6. Krasnik, M.: DIANA – An object-oriented tool for nonlinear analysis of chemical processes. Forschungsberichte Max-Planck-Institut für Dynamik komplexer technischer Systeme Magdeburg, Band 23, Shaker Verlag, Aachen 2008.
7. V. Svjatnyj, M. Resch, V. Kushnarenko, O. Shcherbakov: Dekomposition der verteilten parallelen Simulationsumgebung. Probleme der Modellierung und rechnergestützten Projektierung. DonNTU-Berichte, №1(10)-2(11). Donezk 2012, 227-234.
8. Modelica – A Unified Object-Oriented Language for Physical Systems Modeling. Language Specification. Version 2.0, 2002.
9. Svjatnyj V.A., Moldovanova O.V., Feldmann L.P.: Parallele Simulationsumgebung für dynamische Netzobjekte mit verteilten Parametern. Tagungsband 18. ASIM-Symposium Simulationstechnik, SCS 2005, 416-421.

10. Svjatnyj V., Beljaev O., Lapko V., Tschepzov O., Hohmann R.: Zur Entwicklungsorganisation des Simulations- und Servicezentrums für die Kohleindustrie. Tagungsband 18. ASIM-Symposium Simulationstechnik, SCS 2005, 554-559.
11. Svjatnyj V., Moldovanova O., Smagin A., Resch M., Keller R., Rabenseifner R.: Virtuelle Simulationsmodelle und ein Devirtualisierungsvorgang für die Entwicklung der parallelen Simulatoren von komplexen dynamischen Systemen. DonNTU-FRTI-Berichte, Reihe "Probleme der Modellierung und rechnergestützten Projektierung", Band 5(116), Donezk 2006, 36–43.
12. Dosta, M.: Dynamic Flowsheet Simulation of Solids Processes and its Application to Fluidized Bed Spray Granulation. Cuvillier Verlag, Göttingen, 2013.
13. Bönisch Th., Alessandrini V.: DEISA overview: projekt status, strategies and perspectives. Sixth DEISA Training Session, HLRS Stuttgart 2008, 1-49.
14. Dürr, F.: Improving the Efficiency of Cloud Infrastructures with Elastic Tandem Machines. In: Proceedings of the 6th IEEE International Conference on Cloud Computing (Cloud 2013), Santa Clara/CA/USA 2013.
15. Adinetz, A., Kraus, J., Pleiter, D.: NVIDIA Application Lab at Jülich, inSiDE, Vol. 11(2013), 97-99.
16. Resch, M.: New HLRS System HERMIT, inSiDE, Vol. 9(2011).
17. Feldmann L.P. Implementierung und Effizienzanalyse von parallelen blockartigen Simulationsalgorithmen für dynamische Systeme mit konzentrierten Parametern. Tagungsband 14. ASIM-Symposium Simulationstechnik in Hamburg 2000, SCS-Europe BVBA, Ghent/Belgium 2000, 241-246.
18. Dmitrieva O.A. Parallele Differenzverfahren fürs Lösen des Cauchy-Problems. Donezk, DonNTU 2011 (ukrainisch).
19. Gogolenko S.Y., Svjatnyj V.A. Architecture aware parallelization of solvers for PDE systems on geometrical graphs. Computer science – Research and Development, Vol. 23(2009), 225-230.
20. Resch, M., Kuschnarenko, V., Svjatnyj, V.: Zur Entwicklung der parallelen Simulatoren für dynamische Netzobjekte mit verteilten Parametern. In: Konferenzband „Simulation-2012“, Kyiv 2012, 26-34.
21. Svjatnyj V., Resch M., Zolotukhina O.: Blockorientierte Simulationssprache der parallelen Simulationstechnik. Моделирование - 2016 – Киев: Институт проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины. – 2016. Пленарный доклад, С. 28-36
22. Shcherbakov O., Polsterer K., Svjatnyj V.: Integration of Distributed Parallel Simulation Environment with Cloud-Infrastructures. Моделирование - 2016 – Киев: Институт проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины. – 2016. С. 249-254

Надійшла до редакції 08.10.2016

В.А. СВЯТНИЙ, В.Г. КУШНАРЕНКО, М. РЕШ, О.М. МИРОШКІН, Ш. ВЕСНЕР

ДВНЗ «Донецький національний технічний університет», м. Покровськ, Україна
Інститут управління інформаційними ресурсами, Університет Ульма, Німеччина
Високопродуктивний обчислювальний центр Університету Штутгарту, Німеччина
ПРОБЛЕМАТИКА ПАРАЛЕЛЬНИХ ТЕХНОЛОГІЙ МОДЕЛЮВАННЯ

Сучасне математичне моделювання розглядається в двох аспектах: як міждисциплінарна самостійна область досліджень з фундаментальними теоретичними основами і методами побудови універсальних апаратно-програмних засобів реалізації моделей та як прикладна наукова дисципліна, яка суттєво допомагає всім предметним областям розробляти математичні моделі й симулятори, проводити модельну підтримку проектів автоматизації технологічних процесів і систем, організувати навчання технологічного персоналу на тренажерних симуляторах, інтегрувати засоби моделювання й автоматизованого проектування. Зростаючі вимоги з боку предметних областей до методів і засобів моделювання стосуються обох названих аспектів, стимулюють застосування паралельних надпродуктивних обчислювальних систем наявних і майбутніх MIMD-архітектур і зумовлюють появу нових теоретичних та практичних проблем моделювання, які автори визначають як проблеми паралельних технологій моделювання (parallel Simulationstechnik, parallel Simulation Technology, ParSimTech).

Подальший ParSimTech-розвиток зосереджується на паралельних блокових вирішувачах рівнянь, паралельних алгоритмах і мовах моделювання, об'єктно орієнтованих ієрархічних і архітектурно-залежних підходах до розпаралелювання, проблемно орієнтованих моделюючих середовищах, а також на промислових і університетських впровадженнях.

Ключові слова: паралельне моделювання, генератор рівнянь, розподілене паралельне моделююче середовище, симулятор, топологічний аналізатор, девіртуалізація, ParSimTech.

V.A. SVYATNYI, V.G. KUSCHNARENKO, M. RESCH, O.M. MIROSHKIN, S. WESNER

Donetsk National Technical University, Pokrovsk, Ukraine

Institute of Information Resource Management, University of Ulm, Germany

The High Performance Computing Center Stuttgart of the University of Stuttgart, Germany

PARALLEL SIMULATION TECHNOLOGY ISSUES

Modern mathematical modeling is considered in two ways: as an interdisciplinary independent area of research with the fundamental theoretical bases and construction methods of universal hardware-software means for models implementation and as an applied scientific discipline, which significantly helps all subject areas to develop mathematical models and simulators, to carry out model support of technological processes and systems automation projects, to organize technical personnel training on simulators, to integrate modeling and computer-aided design means. Increasing requirements from the subject areas to the simulation methods and means applies to both these aspects, stimulate parallel overproductive computing systems of existing and future MIMD-architectures usage and determine new theoretical and practical modeling problems appearance, which the authors define as the problem of parallel simulation technology (Parallele Simulationstechnik, ParSimTech), namely:

- Effective (from simulation point of view) parallel hardware-software resources usage: proposed concept of universal and problem-oriented parallel modeling environments and their implementation are a new organization system of parallel tools and modeling techniques.
- Increasing user friendliness of parallel modeling tools: developed topological analyzers and equations generators for dynamic network objects can be implemented in a variety of subject areas.
- Computer Support of models, parallel Simulation-models and simulation design for dynamic systems with lumped and distributed parameters: due to using of the proposed virtual Simulation-models and devirtualization processes the systematization of development stages is achieved.
- Development of parallel algorithms for dynamic systems with lumped and distributed parameters modeling: developed equations solvers on the basis of serial and parallel computing methods are important components of the parallel simulation environments.
- Rapid development and quick implementation of problem-oriented parallel modeling environments: proposed and studied decomposition of parallel simulation environments provides simultaneous system and software-technical development of their functional subsystems.
- Development of parallel simulation languages as an important factor for extending the application of parallel computing systems and design modeling support: the proposed approaches allow gradual transition from the existing standardized programming languages and libraries to high-level languages for model design and carrying out of experiments.
- The integration of methods and tools of parallel simulation with theory and practice of distributed computing systems and modern IT-solutions: the experience of geographically distributed parallel simulation environments implementation and efficient remote access to various parallel resources became a basis for experimental research in the direction of distributed parallel simulation environments with GRID- and CLOUD-infrastructure integration.

Further ParSimTech-development focuses on the block parallel equations solvers, parallel algorithms and modeling languages, object-oriented and hierarchical architecture-specific approaches to parallelization, domain-specific modeling environments and industrial and university implementation as well.

Key words: *parallel simulation, equations generator, distributed parallel simulation environment, simulator, topological analyzer, devirtualization, ParSimTech.*