Национальная академия наук Украины

Институт проблем моделирования в энергетике им. Г.Е. Пухова

МОДЕЛИРОВАНИЕ-2016

SIMULATION-2016

25-27 мая 2016, Киев

Институт проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины
НИИ многопроцессорных вычислительных систем им А.В. Каляева Южного Федерального Университета, Россия
Национальный технический университет «Львовская политехника»
Институт кибернетики им.В.М. Глушкова НАН Украины
Национальный авиационный университет Украины
Донецкий национальный технический университет
Институт электродинамики НАН Украины
Институт проблем регистрации информации НАН Украины
Щецинский технический университет, Польша
Институт специальной связи и защиты информации НТУУ «КПИ»
Ташкентский государственный технический университет, Узбекистан
Компания «Юстар», Украина
Энергосервисная компания «ПАТРИОТ-НРГ»

Сборник трудов конференции

МОДЕЛИРОВАНИЕ-2016

SIMULATION-2016

25-27 мая 2016, Киев

УДК 004.94

М74

Институт проблем моделирования в энергетике им. Г.Е. Пухова НАН Украины, НИИ многопроцессорных вычислительных систем им А.В. Каляева Южного Федерального Университета (Россия), Национальный технический университет «Львовская политехника», Институт кибернетики им.В.М. Глушкова НАН Украины, Национальный авиационный университет Украины, Донецкий национальный технический университет, Институт электродинамики НАН Украины, Институт проблем регистрации информации НАН Украины, Щецинский технический университет (Польша), Институт специальной связи и защиты информации НТУУ «КПИ», Ташкентский государственный технический университет (Узбекистан), Компания «Юстар» (Украина), Энергосервисная компания «ПАТРИОТ-НРГ» проводят Пятую международную конференцию МОДЕЛИРОВАНИЕ-2016.

Конференция проходит в Институте проблем моделирования в энергетике
им. Г.Е. Пухова НАН Украины, г. Киев, 25-27 мая 2016 г.

ISBN 978-966-02-7928-5

Все материалы поданы в авторской редакции. Прошли рецензирование.

ПРОГРАММНЫЙ КОМИТЕТ

Евдокимов Виктор Федорович - сопредседатель (Украина)
Мохор Владимир Владимирович – сопредседатель (Украина)

Аристов Василий Васильевич (Украина)
Белецкий Владимир Николаевич (Польша)
Борукаев Зелим Харитонович (Украина)
Вадреву Шри Хари Рао (Индия)
Верлань Анатолий Федорович (Украина)
Вилински Антоний (Польша)
Винничук Степан Дмитриевич (Украина)
Гнеденко Валерий Герасимович (Россия)
Жуков Игорь Анатольевич (Украина)
Каляев Игорь Анатольевич (Россия)
Катков Александр Федорович (Польша)
Качерек Тадеуш (Польша)
Кириленко Александр Васильевич (Украина)
Макаров Алексей Александрович (Россия)
Массель Людмила Васильевна (Россия)
Осовский Станислав (Польша)
Палагин Александр Васильевич (Украина)
Петров Вячеслав Васильевич (Украина)
Рэш Михаэль (Германия)
Сагатов Миразиз Варисович (Узбекистан)
Самойлов Виктор Дмитриевич (Украина)
Саух Сергей Евгеньевич (Украина)
Святный Владимир Андреевич (Украина)
Семагина Эвелина Петровна (Украина)
Стахив Петр Григорьевич (Украина)
Стогний Борис Сергеевич (Украина)
Тарасенко Владимир Петрович (Украина)
Шидловский Анатолий Корнеевич (Украина)

Оргкомитет конференции «МОДЕЛИРОВАНИЕ-2016»
Председатель Чемерис А.А.
Зам. председателя Клименко Т.М.
Адрес:
Институт проблем моделирования в энергетике
им. Г.Е. Пухова НАН Украины
03164, г. Киев, ул. Генерала Наумова, 15

тел.: (044) 424-9179, (044) 424-1063
факс: (044) 424-0586
e-mail: ipme@ipme.kiev.ua
http://www.ipme.kiev.ua/rus/conference/pukhov100.htm

V.A. Svyatnyy[1)], Prof.
M. Resch[2)], Dr.Sci.Tech
O.A. Zolotukhina[1)]
[1)] Donezker Nationale Technische Universität,
(Shybankov Platz, 2, 85300 Krasnoarmijsk, Ukraine
email: vsvjatnyj@gmail.com)
[2)] Höchstleistungsrechenzentrum
(Stuttgart, Nobelstr. 19, 70569 Stuttgart, Deutschland
email: resch@hlrs.de)

Blockorientierte Simulationssprache der parallelen Simulationstechnik

Das Forschungsgebiet der parallelen Simulationstechnik wird im Rahmen einer langjährigen Kooperation von Informatikern und Simulationstechnikern der Donezkern Nationalen Technischen Universität und der Universität Stuttgart untersucht. Im Mittelpunkt steht dabei die Entwicklung einer leistungsfähigen und benutzerfreundlichen parallelen Simulationsumgebung mit der vollfunktionellen Modellierungs- und Simulationssoftware. In dem Beitrag wird ein Konzept der parallelen blockorientierten Simulationssprache vorgestellt..

Keywords: Simulationssprache, parallelen Simulationstechnik

1. Motivation Die Benutzerfreundlichkeit der parallelen Rechnersysteme ist in ersten Reihe mit den vorhandenen Programmiermodelle und Mitteln für Realisierung der parallelen Lösungsalgorithmen verbunden. Die Systeme der 90-zigen Jahren verfügten über parallele SIMD- und MIMD-Programmiersprachen, die aufgrund von Fortran, C, C++, Modula-2 u. a. Sprachen gebaut wurden [1, 2]. Intensive Entwicklung der MIMD-Systeme und objektorientierten Ansätze hat die Standardisierung der Technologien der parallelen und verteilten Programmierung stimuliert. So wurde ANSI, ISO C++-Standard mit den MPI-, PVM- und Pthreads-Bibliotheken eingeführt. Trotz gewisse Fortschritt soll Gegenstandgebietsexperte (Modellentwickler) heute wie früher mit der Programmiersprache die parallelen Modelle realisieren und damit auf dem Niveau der ehemaligen konventionellen Simulationssysteme der zweiten und dritten Generation [2] bleiben. Diese Tatsache beschränkt wesentlich eine Nutzung der parallelen Ressourcen von Fachleute, die mit den block- (BO), gleichungs- (GO) und objektorientierten (OO) Simulationssprachen [2, 3] dynamische Systeme modellieren und simulieren. Um VPSU zu den Simulationsmitteln etwa der fünften Generation [2] zu nähern, sollen wir die parallele 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. Analyse zeigt, dass die VTF-, SAS- und DNO-Topologien allgemein in der 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. Hier wird das Entwicklungskonzept für parallele BO-Simulationssprache vorgeschlagen.

2. BO-Lösungsprinzip und die MIMD-Parallelität. Hauptkomponente der blockorientierten Simulationssprache ist ein Funktionsblock (Abb. 2), der

programmtechnisch realisiert wird. Block beinhaltet n Eingänge, einziger Ausgang und n Stellen für Einführung der Koeffizienten. Ausgangsvariable wird als Ergebnis der Operation

$$Y = F\,(X_1, X_2, \ldots, X_n, a_1, a_2, \ldots, a_n, t) \qquad (1)$$

dargestellt. Für die Lösung der gewöhnlichen Differentialgleichungssysteme ist folgende

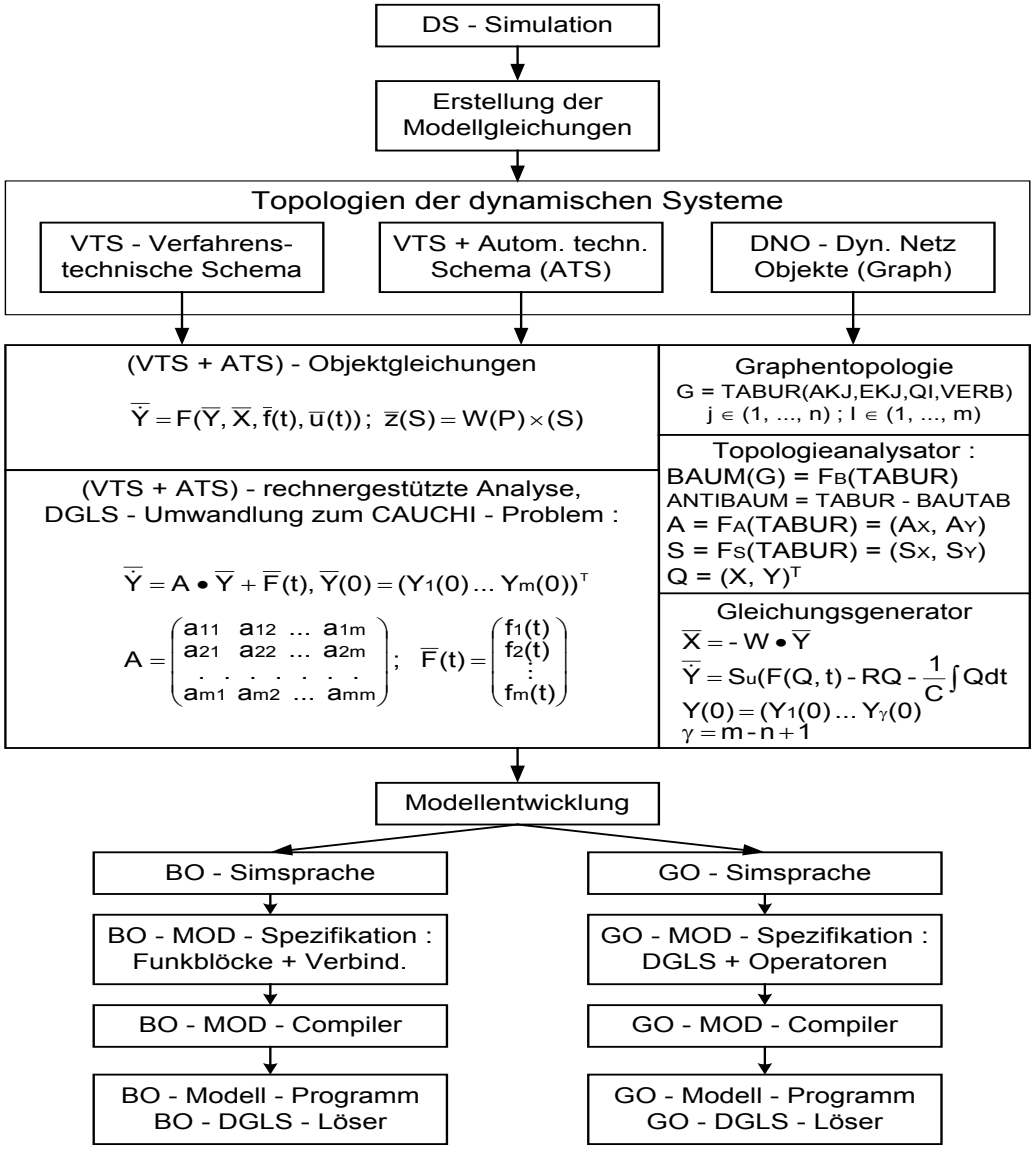

Abb. 1. Modellierung und Simulation von dynamischen Systemen mit den block- und gleichungsorientierten konventionellen Simulationssprachen

Menge der Operationen notwendig

$$F \in \{a_i x_i;\ \sum_{i=1}^{n} a_i x_i; \int \sum_{i=1}^{n} a_i x_i dt; f(x_i), \varphi(x_i, x_k); f(t)\} \qquad (2).$$

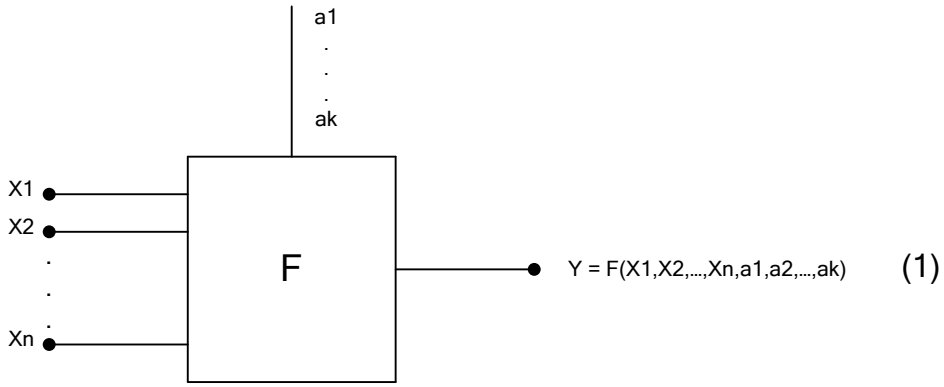

Abb.2. Allgemeine Darstellung des Funktionsblocks einer BO-Simulationssprache

Block-orientiertes (implizites) Modellierungsverfahren wird in eine Reihe von Schritten realisiert:

- ein Modell, z.B.

$$L(dY/dt) + X_1(t)Y|Y| = X_2(Y), \qquad (3)$$

wird zur Form des Simulationsmodells (implizite Funktion bezüglich gesuchtes Y)

$$dY/dt = [X_2(Y) - X_1(t)Y|Y|\ ] / L \qquad (4)$$

umgewandelt;

- mit der Voraussetzung, dass die Variablen $X_2(Y)$ und $- X_1(t)Y|Y|$ der rechten Seite im Simulationsmodell (4) vorhanden sind, werden alle Operationen virtuell mit Hilfe von entsprechenden Funktionsblöcke erfüllt; damit entsteht ein Blockschaltbild der Blöcke, die miteinander entsprechend der Reihenfolge der Operationen in (4) verbunden werden (Abb. 3, Blöcke 1,...,6);
- es werden die Blöcke gesucht, die auf den Ausgängen die Variablen haben, die wir am 1.Schritt als vorhandene vorausgesetzt haben (Block 4 – $X_2(Y)$, Block 6 – $-X_1(t)Y|Y|$, (Abb. 3); mit der Berücksichtigung von +/- Zeichen werden diesen Variablen als Rückkopplungen nach Eingänge des Blocks 1 eingeführt (Abb. 3);

- Blockschaltbild wird auf möglichen semantischen Fehlern geprüft und in PC eingeführt. BO-Compiler wandelt Blocksschaltbild zum lauffähigen BO-Simulator um.

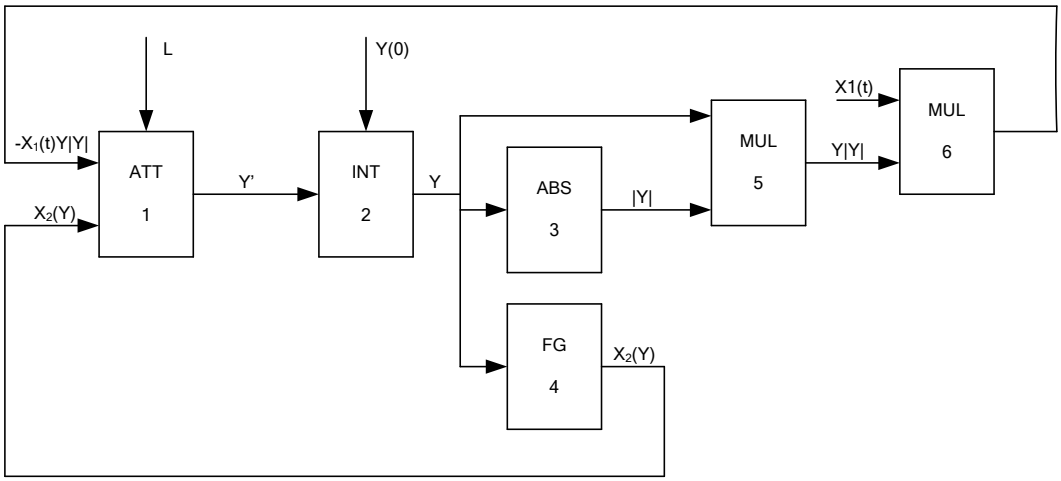

Abb. 3. Blockschaltbild des virtuellen BO-Simulationsmodells

Analyse zeigt, dass BO-Lösungsprinzip entspricht der MIMD-Parallelität und kann als eine virtuelle Zuordnung „Funktionsblock – Prozess“ betrachtet werden (Abb.4).

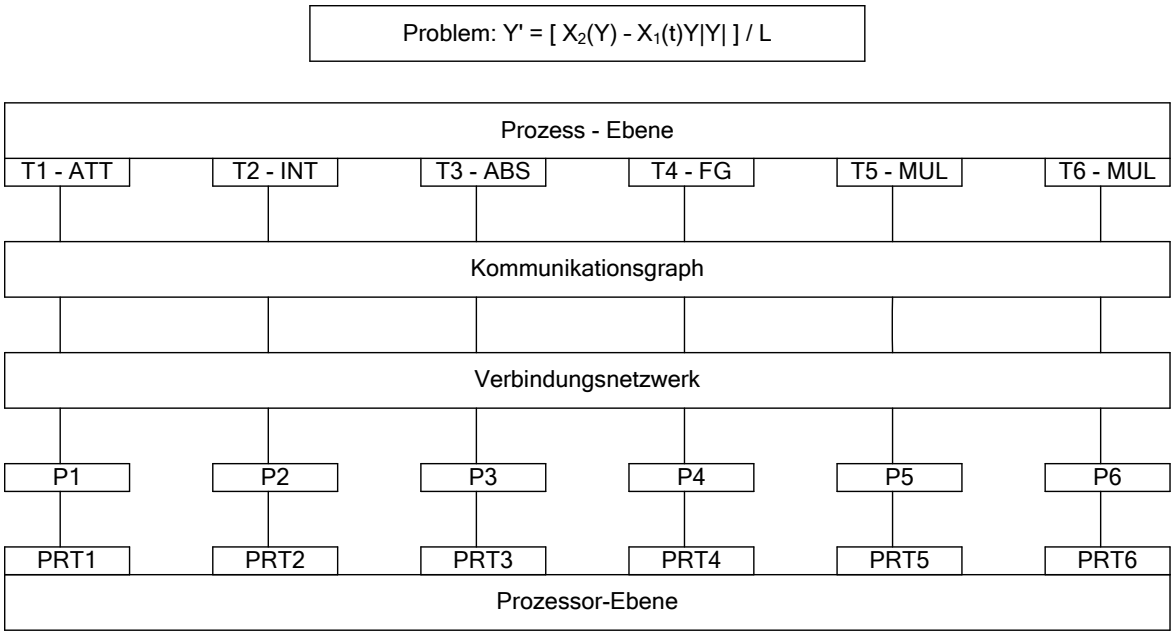

Abb.4 MIMD-Parallelität und BO-Verfahren

3. Definition der virtuellen Paar „Funktionaler Block - Prozess“. Virtueller Funktionsblock wird in Abb. 5 dargestellt. Dabei sind: $\overline{X}$ - Vektor der realen Eingabevariablen; KM – Kommunikationsmatrix; $\overline{VE}$ - Vektor der virtuellen Eingabevariablen. MIMD-Prozess, der dem Funktionsblock entspricht, kann man mit dem Blockdiagramm darstellen (Abb.6).

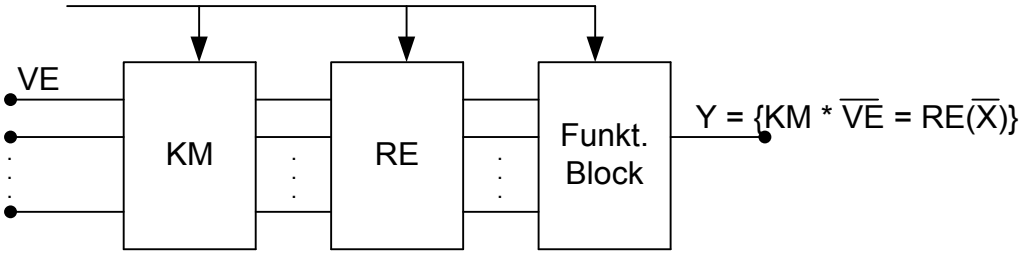

Abb. 5. Virtueller Funktionsblock der BO-Simulationssprache

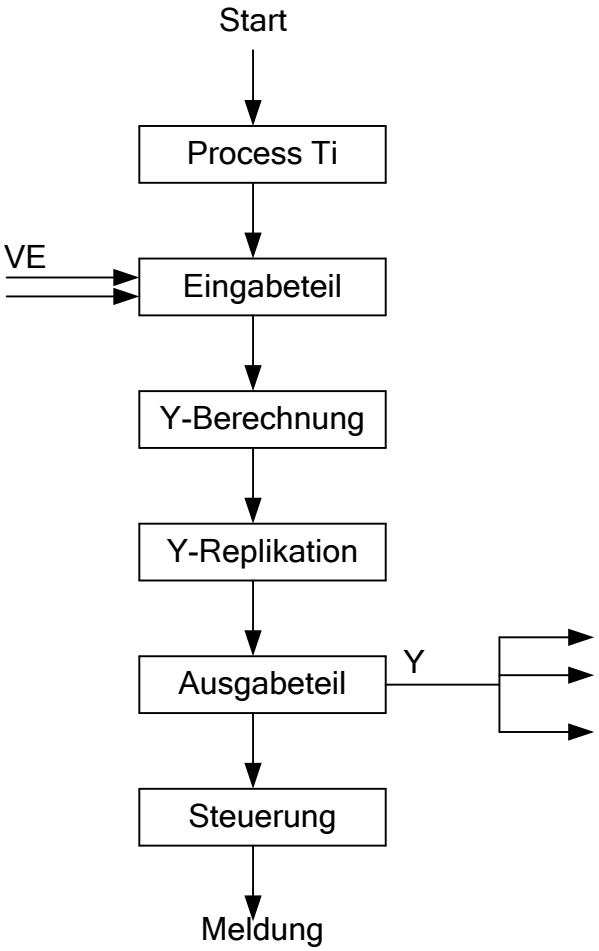

Abb. 6. Blockdiagramm des MIMD-Prozesses.

4. Abbildung der BO-Modellstruktur in virtuelle parallele Modellstruktur (VPMS). VPMS ist eine virtuelle Prozess- und Kommunikationsstruktur. Führen wir Verbindungsvektor für Prozess Ti ein:

$$VSTi = (S_{i1}T_1\ S_{i2}T_2\ \ldots\ S_{ik}\ T_k\ \ldots S_{in}\ T_n) = \bigvee_{\substack{k=1 \\ i \in k}}^{n} (S_{ik}T_k) \qquad (5)$$

Hier sind i – Prozessnummer, i =1,2,…,n; S_{ik} =1 – wenn die Verbindung Ti ↔ Tk vorhanden ist und 0 – wenn gibt es keine Verbindung Ti ↔ Tk.

Für k=i S_{ii} = 1, d.h. Zwischenergebnis wird im i-Prozess für weitere Berechnungen benutzt. Vektor VSTi charakterisiert virtueller Prozess Ti: einziger Ausgang und n Eingänge, die durch Sij bekommen die Variablen von Ausgängen der restlichen Prozesse.

Menge der Verbindungsvektoren für ganze BO-Struktur kann so dargestellt werden

$$\begin{aligned}
&VST_1 = (S_{11}T_1\ S_{12}T_2\ \ldots\ S_{1k}\ T_k\ \ldots S_{1n}\ T_n)\\
&VST_2 = (S_{21}T_1\ S_{22}T2\ \ldots\ S_{2k}\ T_k\ \ldots S_{2n}\ T_n)\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&VST_k = (S_{k1}T_1\ S_{k2}T_2\ \ldots\ S_{kk}\ T_k\ \ldots S_{kn}\ T_n) \qquad (6)\\
&.,\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&VST_n = (S_{n1}T_1\ S_{n2}T_2\ \ldots\ S_{nk}\ T_k\ \ldots S_{nn}\ T_n)
\end{aligned}$$

Mit der Verwendung von (6) kommen wir zur Beschreibung der blockorientierten Struktur des MIMD-Models

$$\begin{aligned}
&VART_1 = FUNT_1(S_{11}VART_1\ S_{12}VART_2\ \ldots\ S_{1n}VART_n)\\
&VART_2 = FUNT_2(S_{21}VART_1\ S_{22}VART_2\ \ldots\ S_{2n}VART_n)\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&VART_k = FUNT_k(S_{k1}VART_1\ S_{k2}VART_2\ \ldots\ S_{kn}VART_n) \qquad (7)\\
&\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\\
&VART_{n-1} = FUNT_{n-1}(S_{n-1,1}VART_1\ S_{n-1,2}VART_2\ \ldots\ S_{n-1,n}VART_n)\\
&VART_n = FUNT_n(S_{n1}VART_1\ S_{n2}VART_2\ \ldots\ S_{nn}VART_n)
\end{aligned}$$

Hier sind VARTi – die Ausgangsvariablen von Prozessen Ti, i = 1,2,...,n; FUNTi – die von Prozessen Ti realisierte Funktionen. Aus (7) folgt die Kommunikationsmatrix

$$KM = \begin{pmatrix} S_{11} & S_{12} & \ldots & S_{1n} \\ S_{21} & S_{22} & \ldots & S_{2n} \\ \ldots & \ldots & \ldots & \ldots \\ S_{k1} & S_{k2} & \ldots & S_{kn} \\ \ldots & \ldots & \ldots & \ldots \\ S_{n1} & S_{22} & \ldots & S_{nn} \end{pmatrix}, \qquad (8)$$

die alle von Blockschaltbild bedingte Verbindungen formal darstellt.

Rechte Seite von (6) lässt sich formal als die Matrix-Multiplikation generieren:

$$VST = \begin{pmatrix} S_{11} & S_{12} & \dots & S_{1n} \\ S_{21} & S_{22} & \dots & S_{2n} \\ \dots & \dots & \dots & \dots \\ S_{k1} & S_{k2} & \dots & S_{kn} \\ \dots & \dots & \dots & \dots \\ S_{n1} & S_{n2} & \dots & S_{nn} \end{pmatrix} * \begin{pmatrix} T_1 & 0 & \dots & 0 \\ 0 & T_2 & \dots & 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 \dots & T_k & \dots\ 0 \\ \dots & \dots & \dots & \dots \\ 0 & 0 & \dots & T_n \end{pmatrix} \quad (9)$$

Die zusammengestellte Vektoren $VST_1 \dots VST_n$ sind die Zeile der Modellzustandsmatrix (MZM):

$$MZM = KM * DT \quad (10)$$

Dabei ist DT eine Diagonalmatrix der T – Prozesse.

5. Devirtualisierung der VPMS. In Abb.7 sind qualitativ die Lastcharakteristiken der Funktionsblöcke gezeigt.

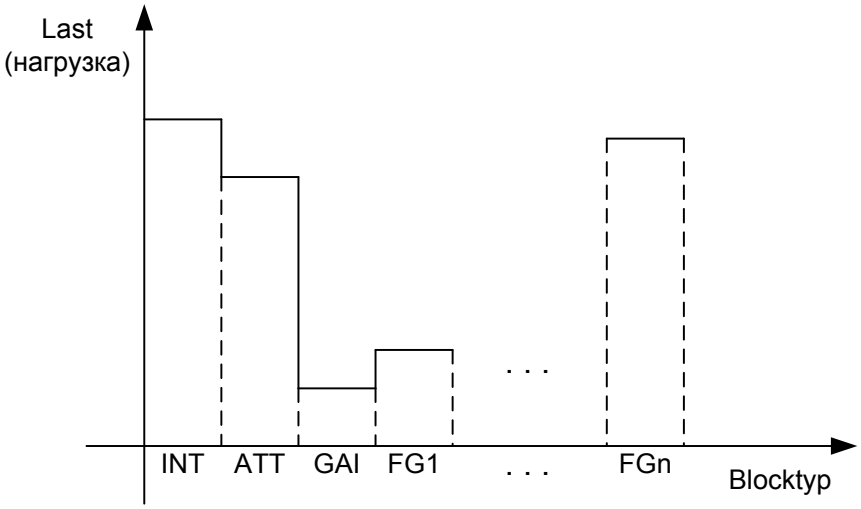

Abb. 7. Zur Lastanalyse der VM-Blockeinheiten

Abhängig von Art der mathematischen Operationen und numerischen Verfahren der Implementierung jeder Block besitzt eigenes Realisierungsprogramm, das für MIMD-Prozess bestimmter Last bedeutet. Prinzipiell ist Struktur der BO-MIMD-Prozesse in Abb.4 lauffähig, aber die Prozesse haben ungleichmäßige Arbeitsumfänge. Deshalb bei der Devirtualisierung soll umfangreiche Lastbalancierung des Blockschaltbildes durchgeführt werden. Dem minimalkörnigen Prozess (MKP) soll eine Funktionsblockkette zugeordnet werden, die eine Differentialgleichung der ersten Ordnung löst. Diesen Ketten werden vom Generator des BO-Lösers nach entwickelte Algorithmen erstellt. So entsteht die von dem BO-Lösungsprinzip bedingte virtuelle Blockschaltung, die das virtuelle parallele

Simulationsmodell der ersten Parallelitätsebene (PE-1) mit minimal körnigen Prozessen „realisiert“. Das virtuelle Verbindungsnetzwerk (VNW) hat zwei Ebene: innere MKP-Verbindungen sowie äußere, die alle MKP entsprechend der Approximationsschema verbinden.

Für die gut strukturierten Simulationsprobleme ist oft möglich eine Matrix-Vektor-Darstellung der Gleichungssysteme zu erstellen, z.B. für dynamisches Netzobjekt

$$X = -WY$$
$$Y' = S_uH - S_uRZ - S_uRR(t)Z \qquad (11)$$

mit folgenden Bezeichnungen: X, Y – Luftstromvektoren in Baum- und Antibaumzweigen; Su – umgewandelte Maschenmatrix; R, RR(t) – diagonale Parametermatrizen; Z – Vektor mit den Komponenten Xj|Xj|,Yk|Yk|, i =1,2, ..,n-1; k = 1,2,..,γ; γ = m-n+1; m, n – die Kanten- und Knotenmenge; H – Vektor der Druckwerten; W – berechnete Topologiematrix.

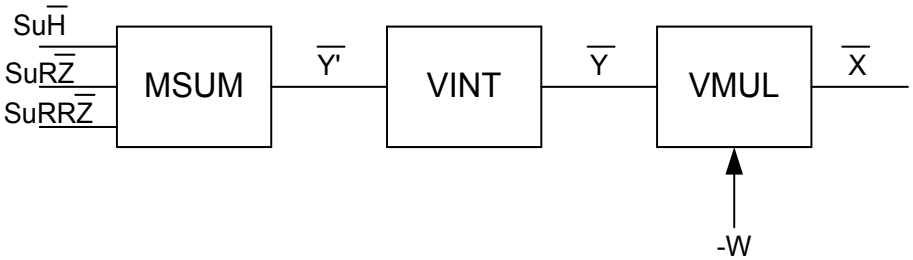

Abb.8. Matrix-Vektor-Funktionsblöcke im Einsatz für das Lösen des Gleichungssystems (11)

Suche der gleichmässigen eindimensionalen und Matrix-Vektor-Lastketten beinhaltet folgende Operationen:

a)Auswahl der lastmaximalen Blöcken(LMB);
b)Lastanalyse der Blöckenketten;
c)Erstellung der Modellstrukture mit den lastgleichen Blockmodule.

Dann wird eine Abbildung der Modellstruktur mit den lastgleichen Blockmodule in Prozessestruktur durchgeführt: Zuordnung “Blockmodule-Prozesse”; Erstellung der Prozessestruktur mit der Definition der Verbindungen zwischen den Prozesse. Struktur der Matrix-Vektor-Prozesse wird ähnlich erstellt. Die Verbindungen zwischen den Prozessen können durch Topologie-und Kommunikationsmatrizen (wie in (11), Matrix W) definiert werden.

Der Devirtualisierungsvorgang soll vom vorgeschlagenen in der VPSU-Simulationssoftware eingebauten BO-Devirtualisator durchgeführt werden. Als Randbedingungen hat er die Parallelitätsebene PE-1 und Zielrechnersystem ZRS. Die optimale Zuordnung dem ZRS wird mit der Berücksichtigung der höheren Parallelitätsebenen (PE) etwa PE-2 – PE-4 und der Abbildung ihren virtuellen Verbindungsnetzwerke VNW-2 – VNW-4 in dem ZRS-Verbindungsnetwerk realisiert. Diese Zuordnung benutzt Code-Generator, der ein lauffähiges MPI-Programm generiert.

6. Zusammenfassung und Ausblick. Die steigenden Anforderungen seitens Gegenstandsgebiete an die Methoden und Mittel der Modellierung und Simulation stimulieren die Anwendung von parallelen Höchstleistungsrechnern der vorhandenen und künftigen MIMD-Architekturen und rufen die neue zu lösenden theoretischen und praktischen Probleme der parallelen Simulationstechnik hervor. Problem der Erhöhung von Benutzerfreundlichkeit paralleler Simulationswerkzeuge wird durch Entwicklung der parallelen Simulationssprachen gelöst sein. BO-Lösensprinzip entspricht der MIMD-Parallelität und kann als eine virtuelle Zuordnung „Funktionsblock – Prozess“ betrachtet werden. Dem minimalkörnigen Prozess soll eine Funktionsblockkette zugeordnet werden, welche eine Differentialgleichung der ersten Ordnung löst. Die betrachteten Ansätze erlauben kontinuierlicher Übergang von vorhandenen standardisierten Sprachen und Bibliotheken der Programmierung zu den höheren Sprachen der parallelen Modellierung und Simulation, sind wichtiger Faktor für die Erweiterung der Anwendung von parallelen Rechnern und der modellgestützten Projektierung.

1. *Feldmann, L.P., Resch, M., Svjatnyj, V.A., Zeitz,M.:* Software-Architektur für parallele Simulationsumgebungen. Plenarvortrag am ASIM‘2014-Symposium Simulationstechnik (Berlin, 03.-05.09.2014), Tagungsband, S.3-7.
2. *Schmidt B.:* Simulationssysteme der 5. Generation. SiP, Heft 1, 1994, S. 5-6.
3. *Maurer, W.:* Modelica simuliert komplexe Systeme. In: Schweizerische Technische Zeitschrift STZ (Swiss Engineering); Nr. 7/8, STV-Verlags AG der Ingenieure und Architekten, Juli/Aug. 2011.

СОДЕРЖАНИЕ

1. *В. В. Петров, А. А. Крючин* Применение оптических сапфировых дисков для хранения стратегически важной информации 7

2. *А.В. Палагин* Ontology conception of transdisciplinary scientific research 11

3. *Oliver Sawodny* Model-Based Optimization methods for operational strategies and layout optimization at the example of Hybrid Energy Systems 18

4. *Volodymyr Kushnarenko, Jürgen Salk, Christian Mosch, Karsten Siegmund, Matthias Neuer, Stefan Kombrink, Thomas Nau, Stefan Wesner* Computational chemistry optimized cluster JUSTUS. Problem oriented approach and architecture. 19

5. *Masha Sosonkina, Gary Lawson, Vaibhav Sundriyal, Yuzhong Shen* Energy Modeling for CoMD Offloaded to Intel Xeon Phi 20

6. *V.A. Svyatnyy, M. Resch, O.A. Zolotukhina* Blockorientierte Simulationssprache der parallelen Simulationstechnik 28

7. *С. Е. Саух* Стабилизированный метод ускоренного решения вариационных неравенств большой размерности 37

8. *К.В. Агеев* Исследование струйно-вихревых процессов в энерготехнологическом оборудовании методами имитационного и численного CFD-моделирования 49

9. *О. А. Дмитриева* О модификации методов типа Биккарта для расширения области устойчивости 53

10. *V. K. Dobrovolsky* Microprocessor with Tagged Registers 57

11. *Е.Е. Федоров, Ю.Л. Дикова* Мультиагентная система прогноза состояния рудничной атмосферы 61

12. *D. Khlopov, M. Mangold* Программное средство для автоматического упрощения моделей дифференциально-алгебраических систем с помощью метода истинного ортогонального разложения 65

13. *В. М. Карпенко* Основні теореми енергоіфнормаційного аналізу динаміки фізичних точок 69

14. *Aleksandr Katkow* Additive Algorithm for Lossy Data Compression 73