

УДК 004.942

## DEKOMPOSITION DER VERTEILTEN PARALLELEN SIMULATIONSUMGEBUNG

V. Svjatnyj<sup>1</sup> (Prof. Dr.-Ing), V. Kushnarenko<sup>1</sup> (Aspirant),  
O. Shcherbakov<sup>1</sup> (Aspirant), Resch<sup>2</sup> (Prof. Dr.-Ing. Dr.h.c.)

<sup>1</sup>Nationale technische Universität Donezk

<sup>2</sup>Höchstleistungsrechenzentrum Stuttgart

[svjatnyj@cs.dgtu.donetsk.ua](mailto:svjatnyj@cs.dgtu.donetsk.ua), [resch@hlrs.de](mailto:resch@hlrs.de)

Verteilte parallele Simulationsumgebung (VPSU) wird als komplexes HW/SW-System betrachtet. Für die Entwicklung der vollfunktionellen Modellierungs- und Simulationssoftware wird eine VPSU-Dekomposition auf Subsysteme vorgeschlagen und die Entwicklungsmethodik für Subsysteme betrachtet.

**Schlüsselwörter:** VPSU, parallele Modellierung und Simulation, Simulationssoftware, funktionelles Subsystem.

### **Einführung**

Die verteilte parallele Simulationsumgebung (VPSU) ist eine benutzerfreundliche simulationstechnische Systemorganisation der parallelen Hardware, der architekturelevanten Systemsoftware, der speziell zielgerichtet entwickelten Modellierungs- und Simulationssoftware, mit denen alle Etappen der parallelen Modellierung und Simulation von komplexen dynamischen Systemen mit konzentrierten (DSKP) und verteilten (DSVP) Parametern unterstützt werden. Die VPSU-Entwicklung wird als zentrales Problem der parallelen Simulationstechnik angesehen [1] und von LCE und HLRS nach der folgenden Konzeption durchgeführt: möglichst volle Berücksichtigung der aktuellen DSKP/DSVP-Anforderungen an die Simulationswerkzeuge; Nutzung der vorhandenen und perspektiven parallelen Ressourcen von MIMD-Architekturen als verteilten Rechensysteme: die zur Zeit den Benutzern zur Verfügung stehende VPSU-Hardware-Struktur zeigt Abb.1; Entwicklung der funktionsvollständigen parallelen Simulationssoftware mit ähnlichen Eigenschaften wie bei den vorhandenen block-, objekt- und gleichungsorientierten Simulationssprachen; möglichst breite Verwendung von denselben Lösensalgorithmen für DSKP und ortsdiskretisierten DSVP; die problemorientierte VPS-Umgebungen werden als Teilprojekte betrachtet, die allgemeine Lösungen nutzen und Ergebnisse für mögliche Verallgemeinerungen liefern; VPSU soll als komplexes HW/SW-SYSTEM mit der umfangreichen Funktionalität entwickelt werden.

In diesem Beitrag wird die VPSU-Dekomposition auf bestimmte Menge der funktionalitätsbedingten Subsysteme als der Hauptansatz zur Entwicklung

der vollfunktionellen Modellierungs- und -Simulationssoftware der VPSU betrachtet.

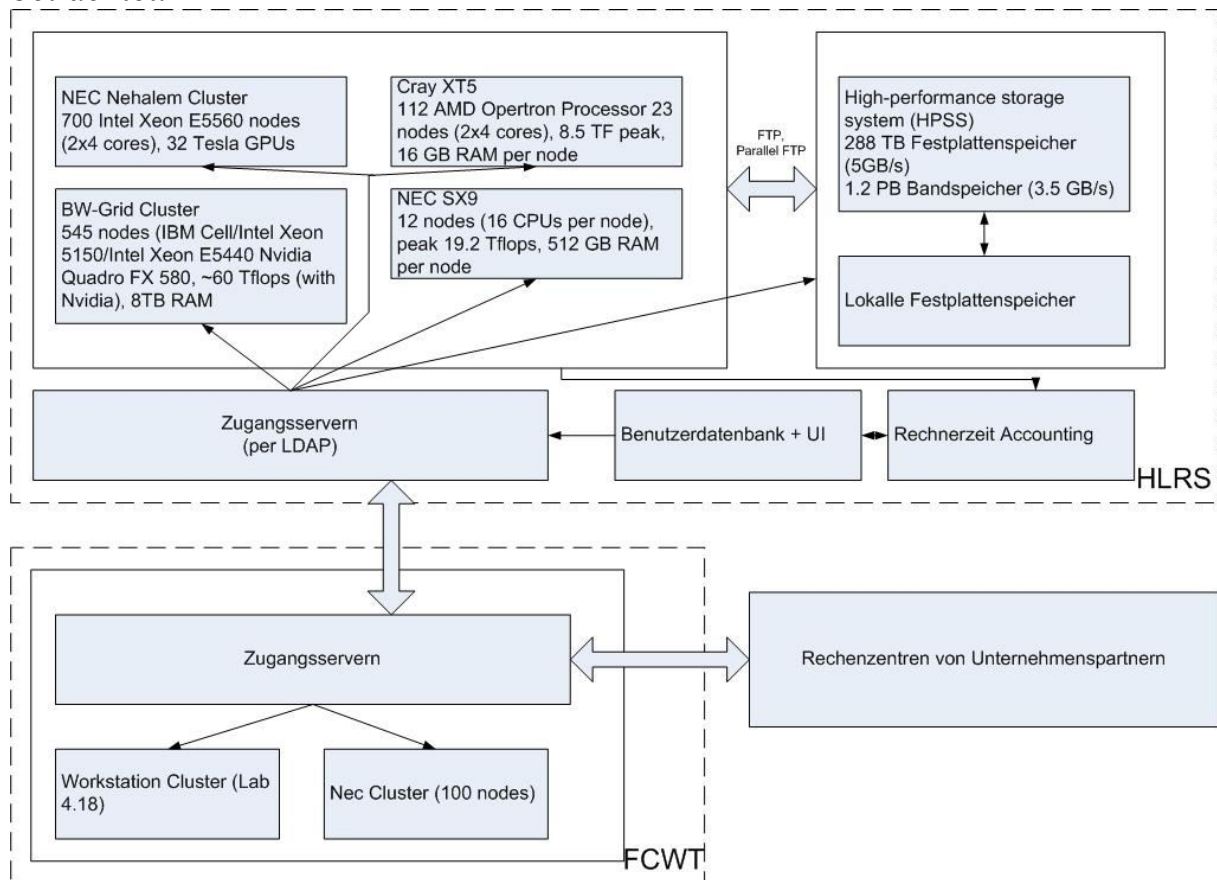


Abb.1. VPSU-Hardwarestruktur.

### **VPSU-Funktionalität**

Die VPSU-Funktionalität, die obiger VPSU-Definition entspricht, wird in Abb. 2 nach drei Funktionsrichtungen verteilt: Funktionen der Modellierung und Simulation von der Problemstellung bis der Bearbeitung von Simulationsergebnissen; Funktionen der Kommunikation von Modellentwicklern und Benutzern mit den VPSU-Ressourcen; Funktionen von Systemorganisation und Infrastruktur.

### **Die VPSU-Subsysteme.**

Es wird vorgeschlagen, das Entwicklungsproblem der vollfunktionellen VPSU-Modellierungs- und Simulationssoftware mit Hilfe der Verteilung der gesamten VPSU-Funktionalität zwischen den Subsystemen zu lösen.

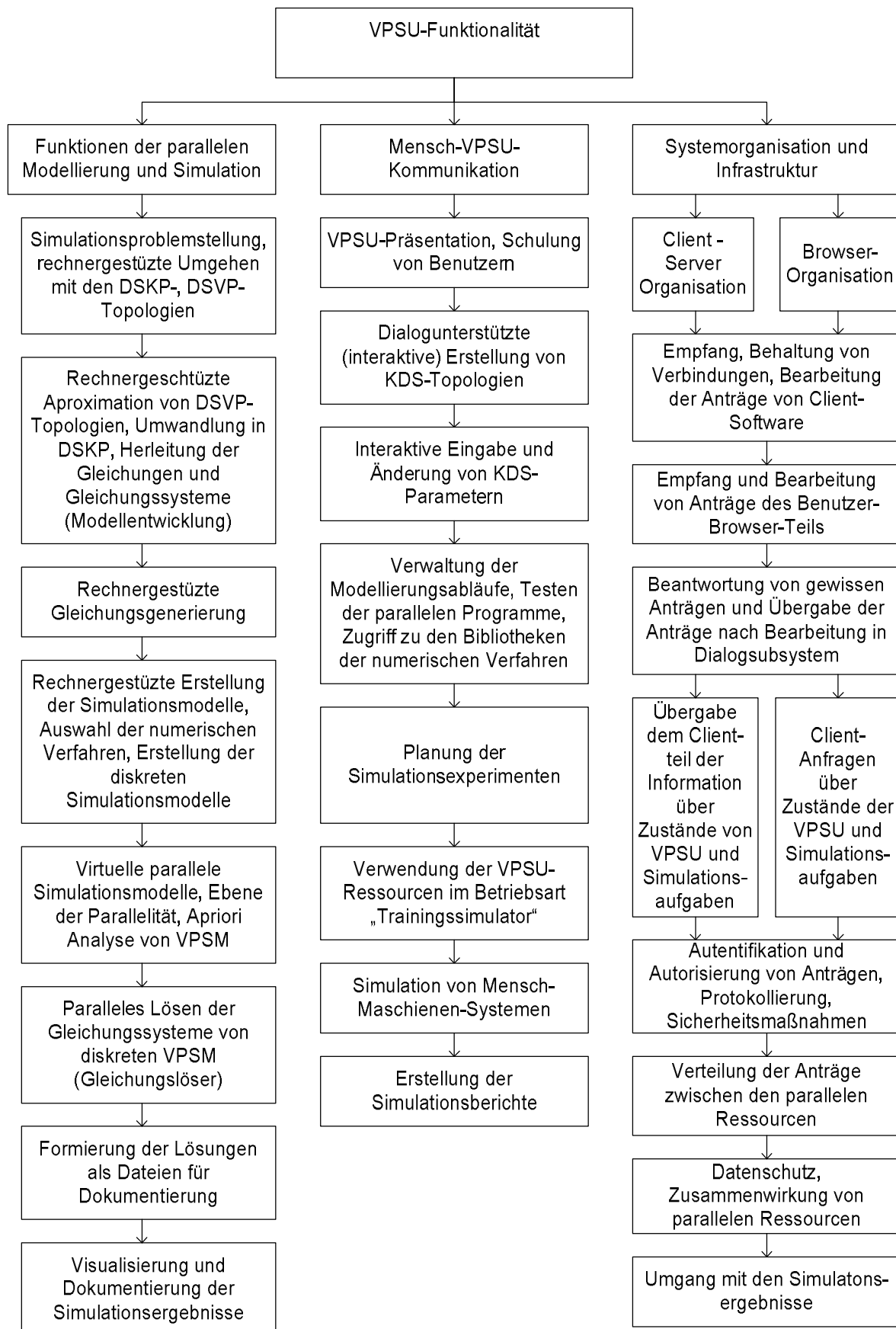


Abb.2. VPSU-Funktionalität und Dekompositionsansatz.

Dabei definieren wir das VPSU-Subsystem als eine Teilkomponente der Hardware, Systemsoftware und Modellierungs/Simulationssoftware, die benutzerfreundlich eine gewisse Gruppe von inhaltlich nahen Funktionen auf entsprechenden parallelen DSKP/DSVP-Modellierungs- und Simulationsetappen realisiert. Nach der Gruppierung und gewisse Detaillierung der in Abb.2 dargestellten VPSU-Funktionen haben wir folgende Liste der Subsysteme mit ihren Hauptfunktionen entwickelt:

1. **Dialogsystem** (DiSuS, die Benutzeroberfläche - BOF) – eine Präsentation der VPSU; Schulung von Benutzern; aktives Dialog der Modellentwickler mit alle Ressourcen des verteilten Rechnersystems; Integration mit alle anderen Subsysteme; Anpassung an die Simulationaufgaben der Gegenstandsgebiete; Planung und Durchführung der Simulationsexperimente.

2. **Topologieanalyse** (TASuS) – verbale und grafische Darstellung sowie geeignete Kodierung von ursprünglichen DSKP/DSVP-Topologien; Dekomposition und Approximation, 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 von Ergebnissen der 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 Multivektor-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 Topologie entsprechend den VPSM-Ebenen; SuGG-unterstützte Formierung der Gleichungssysteme auf alle VPSM-Ebene; Apriori-Analyse der diskreten VPSM Aller Ebenen; interaktive Bereitstellung von implementierungsfähigen diskreten VPSM.

5. **Subsystem von parallelen Gleichungslöser** (SuPaGLö) – Kommunikation mit 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.

6. **Datenaustauschsubsystem (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)** – in Integration mit BOF und Subsystem von 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 DSKP/DSVP von VTF-, SAS- und DNO- Topologien; Archivierung 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.

## **Allgemeine Methodik der Entwicklung von VPSU-Subsystemen**

Dekomposition des komplexen Systems auf bestimmte Menge der Subsysteme ist ein wertvoller Entwicklungsansatz, der den Entwicklern erlaubt, komplexes Problem als Teillösungen darzustellen, ohne gesamte Systemeigenschaften und Funktionen zu verpassen oder nicht zu berücksichtigen. Dabei spielen eine wichtige Rolle neben der Dekomposition selbst die Prozesse der Entwicklungscoordination, des Projektmanagements und der Komposition der Einheiten in den höheren Ebenen der Parallelität aus erhaltenen Dekompositions-lösungen. Es wird von uns VPSU als komplexes HW/SW-System betrachtet und folgende Entwicklungsmethodik vorgeschlagen.

1. **Problemstellung** – wird auf der Ebene des Projektmanagements formuliert. Es sind die oben genannten 10 VPSU-Subsysteme zu entwickeln, zu implementieren und auf den Testsimulationsaufgaben im Rahmen der VPSU-Hardwarestruktur (Abb. 1) experimentell zu untersuchen. Die Projektlösungen sollen als unabhängige von Hardwareplattform konzipiert werden und erst bei der Implementierung die Charakteristiken von Ressource-Komponenten berücksichtigen.
2. **Spezifikation der Funktionen** des  $j$ -Subsystems ( $j=1,2,\dots,10$ ) – wird vom Entwickler des  $j$ -Subsystems erstellt und mit den Entwicklern von alle anderen Subsysteme sowie mit dem Projektleiter koordiniert. Erstelltes Dokument soll eine verbale Spezifikation der Funktionen, die grafische Darstellung der Funktionszusammenhänge, genaue Definition der Verbindungen mit den anderen Subsystemen sowie die formale Beschreibung der Funktionen beinhalten.
3. **Anforderungen** werden von Subsystementwickler mit der Berücksichtigung der Anforderungen an Simulationsmitteln [1], Subsystemfunktionen sowie VPSU- Entwicklungskonzept formuliert und mit den Entwicklern der anderen Subsysteme sowie mit dem Projektleiter abgestimmt.
4. **Beschreibungen** des OO-Ansatzes zur Entwicklung, des Hardware-Teils, der Systemsoftware-Umgebung sowie der BOF/(GUI)-Subsystemfunktionen.
5. **Entwicklung des Simulationssoftware-Teils** soll beinhalten: Entwicklung und Spezifikation der Funktionsalgorithmen; UML-Diagrammen der Subsystem-Simulationssoftware; Vereinbarungen mit den anderen Subsystemen; Vorbereitung der Funktionstestaufgaben für jede Funktion; Vorbereitung der Subsystem-Testaufgaben für die Prüfung der Subsystems-Simulationssoftware; Implementierung und Prüfung der Funktionsalgorithmen in der gewählten Systemsoftware-Umgebung;

- Implementierung und Prüfung der modellierungs- und simulationsspezifischen BOF/(GUI)-Funktionsalgorithmen; Prüfung der Subsystem-Simulationssoftware aufgrund der Subsystem-Testaufgaben.
6. **Entwicklung der subsystemübergreifenden TEST-Modellierung Simulation-Aufgabe** wird auf der Ebene des Projektmanagements formuliert und zusammen mit alle Subsystemprojektvollziehern erfüllt.
  7. **Implementierung der VPSU-TEST-Aufgabe mit der Verwendung von Subsystem-Simulationssoftware:** Verteilung der VPSU-TEST-Aufgabe zwischen den Subsysteme; eine Definition von Subsysteme-Abhängigkeiten und der subsystemübergreifenden Koordination; Implementierung und Debugging der Subsystemteile in VPSU-TEST-Aufgabe; Implementierung und Debugging der subsystemübergreifenden Koordinations-funktionen im Dialog subsystem; Implementierung und Debugging der Subsystem-abhängigkeiten in jedem Subsystem; **VPSU-TEST-Probe**, experimentelle Untersuchungen.

Die Zusammenhänge zwischen den Subsysteme werden in Abb. 3 dargestellt.

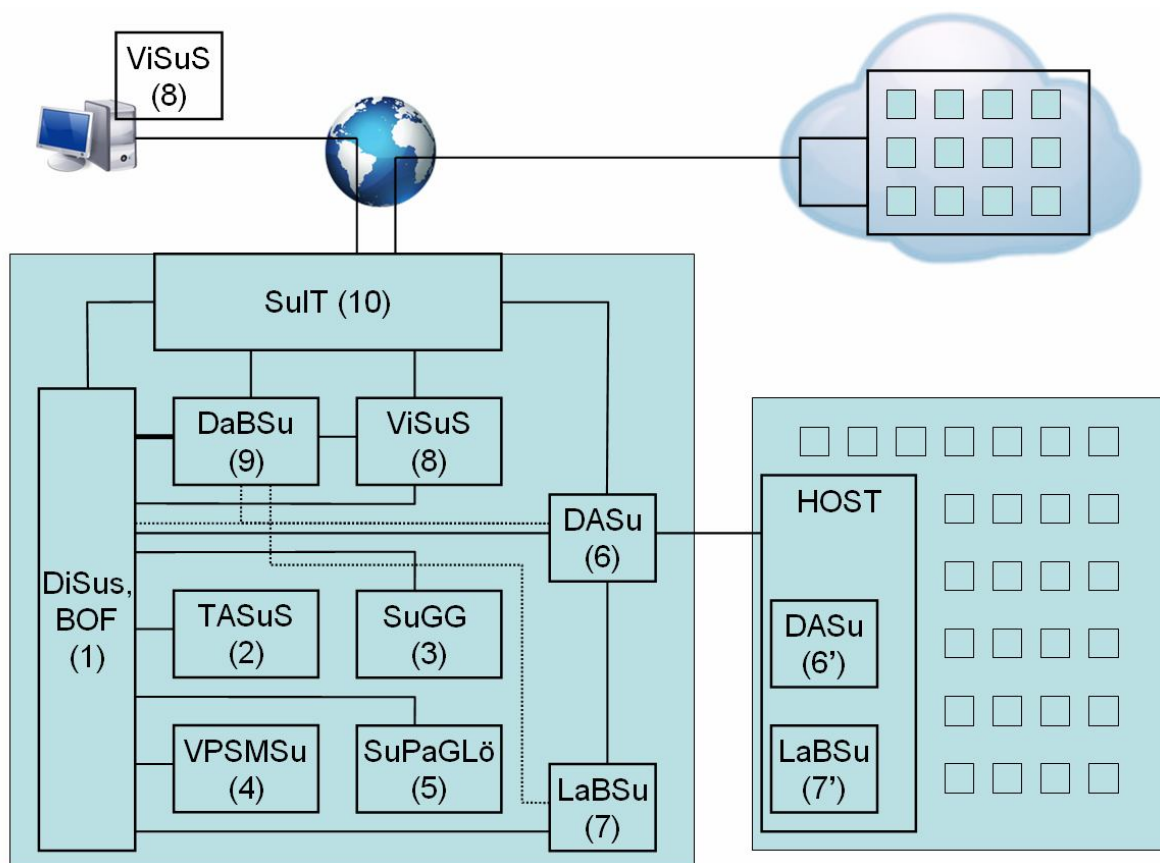


Abb. 3. Schema der Zusammenwirkungen von VPSU-Subsystemen

## **Zusammenfassung**

Die vorgeschlagene Organisation der VPSU-Entwicklung erlaubt nebenläufige Bearbeitung der Teilprojekte (Entwicklungsaufgaben) und eine Erhaltung der Zwischenergebnisse, die zu den arbeitsfähigen Versionen der Simulationsumgebung führen.

## **Literatur**

1. Feldmann L.P., Resch M., Svyatnyj V.A., Zeitz M.: Forschungsgebiet: parallele Simulationstechnik. In: DonNTU, FRTI-Werke, Reihe "Probleme der Modellierung und rechnergestützten Projektierung von dynamischen Systemen", Band 9(150). – Donezk, 2008. – S. 9-36. Auch in: ASIM'09, Tagungsband Vorträgekurzfassungen, Cottbus, 2009.

*Надійшла до редакції 22.09.2012 р.*

*Рецензент: канд. техн. наук, проф. Анопрієнко О.Я.*

**В.А. Святный<sup>1</sup>, В.Г. Кушнарєнко<sup>1</sup>, А.С. Щєрбаков<sup>1</sup>, М. Реш<sup>2</sup>**

<sup>1</sup>Донецкий национальный технический университет

<sup>2</sup>Штутгартский университет

**Декомпозиция распределенной параллельной моделирующей среды на подсистемы.** Распределенная параллельная моделирующая среда (РПМС) рассматривается как сложная аппаратно-программная система. Для разработки полнофункционального моделирующего программного обеспечения предложена декомпозиция среды на подсистемы, рассмотрена методика построения подсистем.

**Ключевые слова:** РПМС, параллельное моделирование, моделирующее программное обеспечение, функциональная подсистема

**В.А. Святный<sup>1</sup>, В.Г. Кушнарєнко<sup>1</sup>, О.С. Щєрбаков<sup>1</sup>, М. Реш<sup>2</sup>**

<sup>1</sup>Донецький національний технічний університет

<sup>2</sup>Штутгартський університет

**Декомпозиція розподіленого паралельного моделюючого середовища на підсистеми.** Розподілене паралельне моделююче середовище (РПМС) розглядається як складна апаратно-програмна система. Для розробки повнофункціонального моделюючого програмного забезпечення запропоновано декомпозицію середовища на підсистеми, розглянуто методику побудови підсистем.

**Ключові слова:** РПМС, паралельне моделювання, моделююче програмне забезпечення, функціональна підсистема