

УДК 004.94:004.8

ВОЗМОЖНОСТИ ПОЛЬЗОВАТЕЛЬСКОГО ИНТЕРФЕЙСА ВИЗУАЛИЗАЦИОННО-ИНТЕГРАЦИОННОЙ ПЛАТФОРМЫ "Оптимус" С УЧЕТОМ ПЕРСПЕКТИВ ИНТЕГРАЦИИ ТЕХНОЛОГИЙ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА

О. В. Коваленко, И. А. Крючков, А. В. Огородников, Н. И. Хочкин
(ФГУП "РФЯЦ-ВНИИЭФ", г. Саров Нижегородской области)

Приводится описание программной архитектуры препостпроцессора визуализационно-интеграционной платформы Оптимус, обеспечивающей оптимизационное имитационное моделирование и управление системами на основе мультиагентных технологий. Данная платформа обеспечивает расширяемый набор компонентов для разработки программных комплексов поддержки принятия решений, востребованных при проектировании и совершенствовании характеристик дорогостоящих беспилотных аппаратов, систем управления, вооружений, военной и специальной техники.

Представлены технологии искусственного интеллекта: предметно-ориентированный язык программирования, онтологическое представление, навигационные сети, метаэвристическая оптимизация. Интеграция этих технологий в графическом пользовательском интерфейсе визуализационно-интеграционной платформы Оптимус позволит существенно расширить спектр возможностей по созданию и совершенствованию вооружений, военной и специальной техники.

Ключевые слова: имитационное моделирование, препостпроцессор, искусственный интеллект, интеллектуальные агенты, предметно-ориентированный язык, онтологии, навигационная сеть, метаэвристическая оптимизация, вооружение, военная и специальная техника, пользовательский интерфейс, визуализационно-интеграционная платформа Оптимус.

Введение

Постоянное совершенствование характеристик вооружений, военной и специальной техники (ВВСТ) — одна из важнейших задач как предприятий ВПК, так и Министерства обороны РФ. Как правило, эта задача требует непрерывного поиска новых современных технических и управленческих решений.

Актуальные тенденции развития информационных технологий делают доступными все больше и больше различных инструментальных средств с элементами искусственного интеллекта (ИИ): нейросетевые программные библиотеки, библиотеки машинного обучения, библиотеки генетического программирования, программные продукты создания и анализа онтологий, различные базы знаний, методы и алгоритмы метаэвристической оптимизации, платформы мульт-

агентного моделирования [1] и др. Указанные технологии ИИ могут быть использованы для расширения спектра возможностей по совершенствованию ВВСТ.

Области применения подобных технологий весьма разнообразны. При проектировании и разработке ВВСТ они позволят разработчикам быстро находить оптимальные, эффективные и нестандартные решения. Также элементы ИИ могут быть использованы непосредственно в образцах вооружений, обеспечивая им уникальные возможности и преимущества (таких вооружений становится все больше). Другим вариантом применения технологий ИИ является создание и использование программных комплексов имитационного моделирования и поддержки принятия решений. Такие комплексы применимы на всех уровнях планирования: от разработки тактики применения отдельного образца вооружений до

планирования и анализа стратегических операций, включающих десятки и сотни тысяч разрозненных объектов.

В каждой из перечисленных задач существенной проблемой является эффективное объединение отдельных технологий ИИ. Интеграция разноплановых инструментов ИИ требует продуманности как на логико-математическом уровне, так и на уровне программных и визуальных интерфейсов.

С целью практического развития данных направлений авторами ведется разработка визуализационно-интеграционной платформы (ВИП) для оптимизационного имитационного моделирования и управления системами (ОптИМУС) [2]. В программную архитектуру данной платформы изначально заложены подходы, позволяющие на разных этапах эксплуатации и разработки использовать ряд технологий ИИ. ВИП ОптИМУС представляет собой расширяемый набор компонентов для разработки программных комплексов имитационного мультиагентного моделирования и включает в себя два программных пакета (ПП):

- мультиагентного моделирования Оптимус.Решатель;
- препостпроцессинга (препостпроцессор) мультиагентных задач Оптимус.Препост с графическим пользовательским интерфейсом (ГПИ).

Далее рассмотрим программную архитектуру и возможности ВИП ОптИМУС, особенно в ча-

сти ГПИ, а также перспективы интеграции в платформу новых передовых технологий.

Программная архитектура препостпроцессора

Краткая схема функционирования ВИП ОптИМУС представлена на рис. 1.

Как видно, ПП Оптимус.Препост (далее препостпроцессор) предназначен для выполнения следующих задач:

- подготовка исходного сценария задачи;
- настройка оптимизационной задачи;
- запуск расчета;
- отображение и анализ результатов расчета.

Препостпроцессор разработан на языке программирования C++. При разработке использованы свободно-распространяемые программные библиотеки, среди которых можно выделить:

- Delta3D — игровой движок с возможностью редактирования игровых сценариев как библиотека базовой функциональности ГПИ;
- osgEarth — библиотека виртуального глобуса как геопространственное средство визуализации моделируемого сценария на карте.

Препостпроцессор разработан как кроссплатформенное приложение и поддерживает работу в операционных системах семейств Windows и Linux. На рис. 2 представлен ГПИ открытого

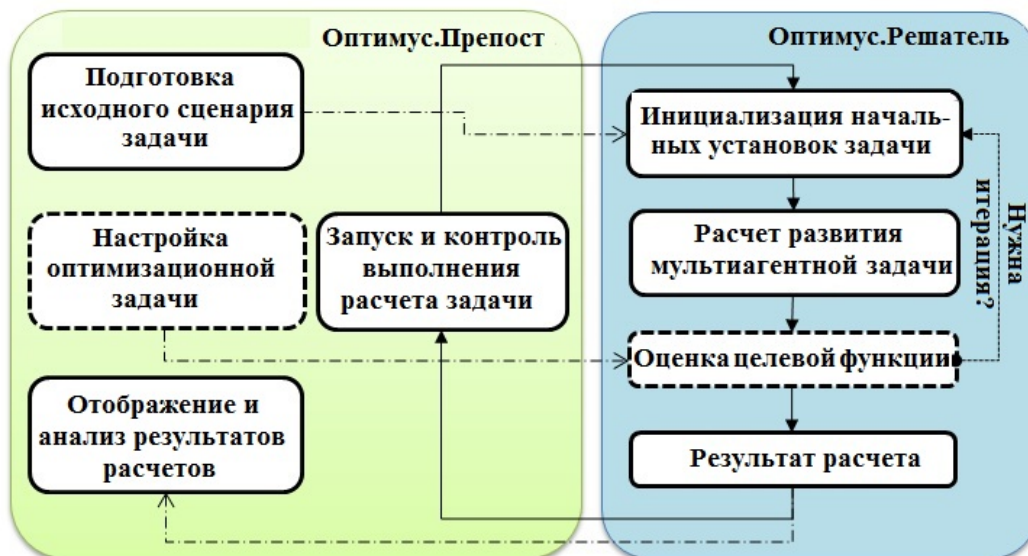


Рис. 1. Краткая схема функционирования ВИП ОптИМУС

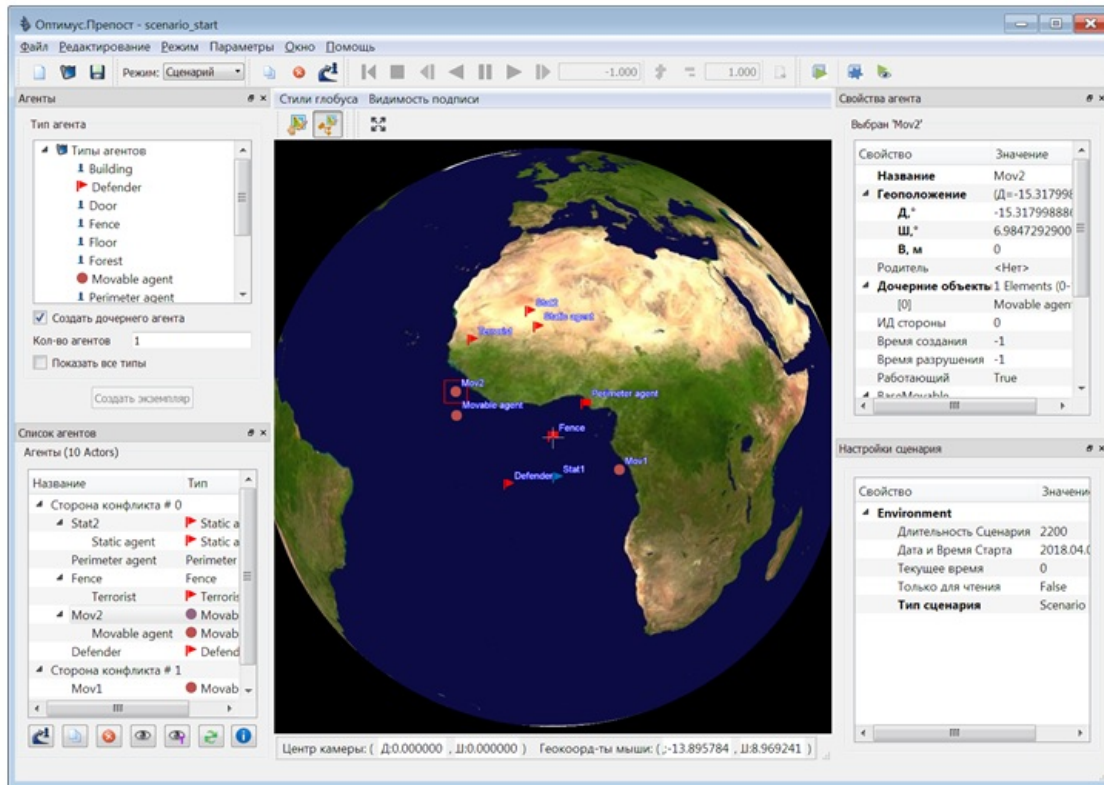


Рис. 2. Пример открытого окна препостпроцессора

окна препостпроцессора с тестовым набором агентов.

Модульная архитектура препостпроцессора предполагает задействование набора базовых модулей разной функциональности, объединенных в составе единого ГПИ. Интеграция модулей препостпроцессора осуществляется средствами программной библиотеки Qt в виде однодокументного оконного интерфейса с набором вложенных окон (док-виджетов).

Схема взаимодействия базовых модулей в составе препостпроцессора между собой и с внешними программными средствами представлена на рис. 3.

Основную функциональность препостпроцессора можно разделить на две части: подготовка исходных сценариев к расчету (препроцессинг); отображение и анализ результатов расчета (постпроцессинг).

Для расширения функциональности базовой платформы препостпроцессор обладает механизмом динамического подключения плагинов. Предназначение плагинов может быть разным — от подключения предметной библиотеки агентов до интеграции диалогового окна дополнительного анализа результатов расчета.

Концепция ВИП Оптимус предполагает работу со сценариями, представляющими собой сконфигурированную для расчета совокупность агентов, вовлеченных в конкретную задачу имитационного моделирования. Сценарии могут быть, как минимум, двух видов:

- 1) исходный (подготовленный в ПП Оптимус.Препост);
- 2) результирующий (полученный в результате расчета ПП Оптимус.Решатель).

Сценарии могут храниться в виде файлов собственного формата в файловой системе или в виде специального представления в поддерживаемой БД PostgreSQL для обеспечения многопользовательского режима работы.

При подготовке исходных сценариев моделирования базовая функциональность препостпроцессора обеспечивает следующие возможности:

- создание и размещение экземпляров заданного типа агента;
- отображение и изменение параметров агентов;
- запуск расчета и просмотр статуса задач;



Рис. 3. Схема взаимодействия модулей препостпроцессора

- сохранение и открытие сценариев, настройка программы.

С целью обеспечения качественной геоинформационной поддержки препостпроцессор визуализирует картографическую информацию с помощью встроенного модуля виртуального глобуса, подключенного к серверу геоинформационной системы (ГИС) по стандартному протоколу WMS/WFS. Виртуальный глобус обеспечивает интерактивное отображение карт на трехмерном геоиде/эллипсоиде WGS84 или ПЗ-90 либо в проекции на плоскость (например, проекция Меркатора). Визуальные агенты отображаются на геосцене виртуального глобуса в соответствии с сопоставленным им способом визуализации (точечной, иконочной, площадной, объемной, 3D-модельной).

Запуск расчета задачи осуществляется по окончании подготовки исходного сценария в ГПИ препостпроцессора. Запуски расчета могут быть как локальными, так и удаленными на доступной в локальной вычислительной сети суперЭВМ, на которой предварительно установлен ПП Оптимус.Решатель.

По окончании расчета анализ в препостпроцессоре предусматривает анимированный интерактивный просмотр результатов моделирования на виртуальном глобусе с возможностью отслеживания таблицы изменяющихся свойств агентов в процессе продвижения по модельному времени.

Также при анализе результатов расчета используются модуль статистики поведения агентов и модуль протокола событий, предоставляющие данные в табличном виде. В статистическом виде могут быть представлены произвольные количественные и суммарные параметры: эффективность работы, количество выполненных рейсов, количество выживших в бою и т. п. Протокол событий показывает упорядоченные по шкале модельного времени записи, состоящие из времени, типа события, участвующих агентов и при необходимости дополнительной информации.

Интеграция технологий ИИ на уровне ГПИ

Современные тенденции оказывают влияние на развитие ВИП Оптимус по направлению использования перспективных технологий ИИ. В данный момент прорабатывается внедрение следующих элементов ИИ на уровне ГПИ:

- 1) программирование логики поведения интеллектуальных агентов с использованием предметно-ориентированных языков;
- 2) онтологические механизмы описания предметной области, включая дескрипционную и нечеткую логику;
- 3) биоинспирированные методы метаэвристической оптимизации;
- 4) навигационные сети для построения маршрутов.

Кроме того, в процессе развития платформы предполагается интеграция еще двух важных технологий ИИ, рассмотрение которых остается за пределами данной статьи:

- 5) графо-вероятностные методы описания логики принятия решений;
- 6) машинное обучение для имитации поведения интеллектуального агента.

Далее рассмотрим первые четыре технологии ИИ и предполагаемые возможности по их использованию в составе ГПИ ВИП ОпТИМУС.

Предметно-ориентированный язык программирования логики агента. Одним из самых трудоемких и критически важных процессов при создании мультиагентных систем является разработка функциональных и поведенческих моделей агентов. В зависимости от требований к агентам, уровня проработанности, сложности предметной области и квалификации разработчиков создание агентов может выполняться различными способами. Для этих целей в ВИП ОпТИМУС предполагается использование как классических языков программирования высокого уровня (C/C++), так и специальных предметно-ориентированных языков (ПОЯ) (в англоязычной литературе DSL — Domain-Specific Language) [3].

ПОЯ, как правило, является расширением регулярного языка программирования и при этом за счет своей полноты обеспечивает быструю и эффективную разработку логики принятия решений, включая поведенческую логику, логику выбора оптимального пути или выбора оптимальной конструкции/структуры. Реализация интеллектуального программного кода на ПОЯ существенно упрощает понимание и сопровождение разрабатываемых моделей агентов (рис. 4). Также ПОЯ позволяет внедрять блоки машинного обучения для имитации интеллектуального поведения агента [1] взамен сложных математических расчетов.

Для улучшения визуального восприятия программ на ПОЯ зачастую применяется механизм мультиграфов. Визуальное программирование, представленное на рис. 4, позволяет привлекать к процессу программирования специалистов в исследуемой предметной области, которые, как правило, программистами не являются.

Хотя визуальное восприятие является крайне важной причиной использования ПОЯ, для ИИ

более важным является то, что пространство операторов и состояний программ на ПОЯ имеет значительно меньшую размерность, чем в аналогичной программе, написанной на языке программирования более низкого уровня. Программы на ПОЯ являются *строгими* структурами, что позволяет в пространстве таких структур проводить эффективный поиск необходимых или в определенном смысле оптимальных структур-программ.

Онтологический подход. Для использования формализованных экспертных знаний при разработке шаблонов поведенческой логики создаваемых и исследуемых программных агентов целесообразно применять современные онтологические методы описания предметной области [4].

Онтологический подход предполагает наличие универсальных механизмов создания формализованной семантической сети, описывающей предметную область моделируемого ИИ. Под семантической сетью подразумевается иерархический способ описания терминологической взаимосвязи и классификации экземпляров, понятий, атрибутов и отношений. Универсальность такого подхода обеспечивает возможность адаптации онтологического инструментария для конкретных прикладных задач, например экспертного описания предметной области конкретного рода войск, в которых планируется применение интересующих образцов ВВСТ.

Использование существующих стандартных форматов представления знаний (например, OWL, OWL DL), описывающих понятия предметной области в недвусмысленном, формализованном виде, а также инструментов логического вывода (дескрипционной и нечеткой логики) позволит расширить языковые возможности ПОЯ, а также обеспечит вычислительные возможности автоматического умозаключения в процессе мультиагентного моделирования.

Для обеспечения онтологической функциональности в препостпроцессоре ВИП ОпТИМУС реализуются:

- создание инструментов ввода экспертного знания в предметную базу знаний;
- использование понятий из онтологии в программировании агентов на ПОЯ, например, для классификации агентов и их манипуляторов;

The image displays a software development environment with two main components: a code editor and a visual programming interface.

Code Editor: The top portion shows JavaScript code for handling mouse events and calculating a click position. The code includes comments and logic for converting 3D space coordinates to 2D space.

```

4 # Member variables
5 var prev_pos = null
6 var last_click_pos = null
7 var viewport = null
8
9
10 = func _input(event):
11 # Check if the event is a non-mouse event
12 # var is_mouse_event = false
13 # var mouse_events = [InputMouseButton, InputEventMotion, InputEventScreenDrag, InputEventScreenTouch]
14 # for mouse_event in mouse_events:
15 #     if (event is mouse_event):
16 #         is_mouse_event = true
17 #         break
18 #
19 # If it is, then pass the event to the viewport
20 # if (is_mouse_event == false):
21 #     viewport.input(event)
22
23
24 # Mouse events for Area
25 = func _on_area_input_event(camera, event, click_pos, click_normal, shape_idx):
26 # Use click_pos (click in 3D space, convert to area space)
27 # var pos = get_node('Area').get_global_transform().affine_inverse()
28 # the click pos is not zero, then use it to convert from 3D space to area space
29 # if (click_pos.x != 0 or click_pos.y != 0 or click_pos.z != 0):
30 #     pos = click_pos
31 #     last_click_pos = click_pos
32 # else:
33 #     # otherwise, we have a motion event and need to use our last click pos
34 #     # and move it according to the relative position of the event.
35 #     # NOTE: this is not an exact 1-1 conversion, but it's pretty close
36 #     pos = last_click_pos
37 #     if (event is InputEventMotion or event is InputEventScreenDrag):
38 #         pos.x += event.relative.x / viewport.size.x
39 #         pos.y += event.relative.y / viewport.size.y
40 #     last_click_pos = pos
41
42 # Convert to 2D
43 pos = Vector2(pos.x, pos.y)
44
45 # Comment: For debugging, uncomment the next line
    
```

Visual Programming Interface: The bottom portion shows a node-based workflow. Key nodes include:

- Function:** A node for the `_on_timer_timeout` function.
- Sequence:** A node with two steps: "Get the size of the viewport (root node)" and "Instantiate a circle with random parameters".
- SceneNode /root:** A node representing the root scene node.
- InstanceSet:** A node for managing a set of instances.
- CallInstance:** A node for calling a specific instance.
- randf:** A node for generating random float values.
- new Vector2():** A node for creating a new 2D vector.
- randf_range:** A node for generating random values within a specific range.
- InstanceSet:** A node for setting instance parameters like `Node2D:scale`.
- CallSelf:** A node for calling the current node's function.

The visual programming interface uses a grid background and various colored nodes (red, green, blue, purple) connected by lines to represent the logic flow of the code.

Рис. 4. Пример программирования поведения агентов в прототипированном процессоре

- формирование выражений логического вывода на основе онтологических знаний.

Навигационные сети. Одной из важных пространственных задач, стоящих перед разработчиком логики интеллектуальных агентов, является задача поиска кратчайшего по заданным критериям пути в пространстве [5, 6], которая решается при помощи достаточно быстрых алгоритмов, таких как A^* или алгоритм Дейкстры. При этом важно предоставить разработчику отладочный механизм просмотра проложенного маршрута в каждый момент времени "мышления" агента. Для этих целей в препостпроцессоре создается визуализатор интеллектуальных маршрутов.

Кроме того, в качестве поддержки интеллектуальной навигации в процессе имитационного моделирования для восстановления геометрического пространства мира разрабатывается возможность генерации экземпляров агентов и самих типов агентов из источников данных:

- баз знаний, содержащих онтологическое описание моделируемого мира;
- ГИС, обеспечивающих картографическую и геометрическую информацию об инфраструктурных и природных объектах.

Метаэвристическая оптимизация. Важным классом задач, решаемых в ВИП, являются оптимизационные задачи. Для их решения в ВИП ОпТИМУС используются биоинспирированные метаэвристические методы [7], основанные на имитации естественных процессов, заимствованных у живой природы, и реализующие адаптивный случайный поиск. Данное направление математики активно развивается. По разным признакам выделяют алгоритмы: популяционные и непопуляционные; натуральные и ненатуральные; эволюционные, роевые и иммунные; непрерывные, дискретные и дискретно-непрерывные. Библиотека алгоритмов в ВИП постоянно расширяется.

Оптимизационные алгоритмы могут использоваться на различных стадиях решения задачи: с одной стороны, в процессе мультиагентного расчета отдельными агентами для поиска локальных оптимальных поведений, с другой стороны, при поиске оптимальных решений для всей системы. Такие разные постановки требуют от препостпроцессора (и ПОЯ, в частности)

специфических возможностей по заданию оптимизационных параметров, определению целевой функции, управлению стохастическими параметрами, анализу полученных результатов.

Заключение

Рассмотренные в статье возможности ВИП ОпТИМУС и перспективы интеграции ряда технологий ИИ в платформу имитационного моделирования позволяют вывести на новый уровень качество проектирования сложных систем, таких, например, как ВВСТ. Инструментарий программных комплексов, создаваемых на базе развиваемой ВИП ОпТИМУС, может стать важной поддержкой при принятии решений по обоснованию тактико-технических характеристик перспективных изделий военно-промышленного комплекса, обеспечивая при этом эффективную и качественную разработку интеллектуальных моделей автономных аппаратов и имитацию их применения в виртуальном геопространстве.

Список литературы

1. *Russel S., Norvig P.* Искусственный интеллект: современный подход, 2-е изд.: Пер. с англ. М.: Вильямс, 2006.
Rassel S., Norvig P. Iskusstvenny intellekt: sovremennyy podkhod. 2-e izd.: Per. s angl. M.: Vilyams, 2006.
2. *Коваленко О. В., Крючков И. А., Ежов Д. В., Огородников А. В., Ерошкина И. В., Собанин Д. С., Хочкин Н. И., Варгина Е. Ф., Тихомиров Ю. В., Рыжик А. В., Васильева Е. А., Кондратьев А. Б.* Визуализационно-интеграционная платформа для оптимизационного имитационного моделирования и управления системами (ВИП ОпТИМУС). Свидетельство о государственной регистрации программы для ЭВМ № 2019617157.
Kovalenko O. V., Kryuchkov I. A., Ezhov D. V., Ogorodnikov A. V., Eroshkina I. V., Sobanin D. S., Khochkin N. I., Vargina E. F., Tikhomirov Yu. V., Ryzhikh A. V., Vasileva E. A., Kondratev A. B. Vizualizatsionno-integratsionnaya platforma dlya optimizatsionnogo imitatsionnogo modelirovaniya i upravleniya sistemami (VIP OptIMUS). Svidetelstvo o gosudarstvennoy

- registratsii programmy dlya EVM № 2019617157.
3. *Bordini R. H., Dastani M., Dix J., Seghrouchni A. El F.* Multi-Agent Programming. Languages. Tools and Applications. N.-Y.: Springer Science+Business Media, LLC, 2009.
 4. *Каширин Д. И., Каширин И. Ю.* Модели представления знаний в системах ИИ // Вестник РГРТУ. 2010. № 1 (вып. 31). С. 36–43.
Kashirin D. I., Kashirin I. Yu. Modeli predstavleniya znaniy v sistemakh II // Vestnik RGRTU. 2010. № 1 (vyp. 31). S. 36–43.
 5. *Hart P. E., Nilsson N. J., Raphael B. A.* Formal basis for the heuristic determination of minimum cost paths // IEEE Transactions on Systems Science and Cybernetics. 1968. Vol. 4 (2). P. 100–107.
 6. *Demyen D. J.* Efficient Triangulation-Based Pathfinding. University of Alberta, 2007.
 7. *Скобцов Ю. А., Федоров Е. Е.* Метаэвристики: монография. Донецк: Ноулидж, 2013.
Skobtsov Yu. A., Fedorov E. E. Metaevristiki: monografiya. Donetsk: Noulidzh, 2013.
- Статья поступила в редакцию 14.03.19.

CAPABILITIES OF THE USER INTERFACE OF THE "OptIMUS" VISUALIZATION-AND-INTEGRATION PLATFORM WITH REGARD TO PROSPECTS FOR THE ARTIFICIAL INTELLIGENCE TECHNOLOGIES INTEGRATION / O. V. Kovalenko, I. A. Kryuchkov, A. V. Ogorodnikov, N. I. Khochkin (FSUE "RFNC-VNIIEF", Sarov, N.Novgorod region).

The paper describes the program architecture of the pre-postprocessor of the OptIMUS visualization-and-integration platform for the optimization simulation and management of systems on the base of multiagent technologies. The platform provides a scalable set of components for the development of program complexes supporting the decision-making processes, which are required in designing and improving characteristics of high-cost unmanned aircrafts, control systems, armaments, military and special-purpose equipment.

The artificial intelligence technologies are presented: a domain-specific programming language, ontological representation, navigation networks, metaheuristic optimization. The integration of these technologies in GUI of the OptIMUS visualization-and-integration platform would allow significantly extending the range of capabilities for the development and improvement of weapons, military and special-purpose equipment.

Keywords: simulation, pre-postprocessor, artificial intelligence, intellectual agents, domain-specific language, ontology, navigation network, metaheuristic optimization, armaments, military and special-purpose equipment, user interface, visualization-and-integration platform OptIMUS.
