

А.Ф. Зайцев**ОСОБЕННОСТИ ПРИМЕНЕНИЯ АГЕНТНО-МОДУЛЬНОГО ПОДХОДА ПРИ ПОСТРОЕНИИ И РЕАЛИЗАЦИИ ИНФОРМАЦИОННО-ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ**

Информационно-вычислительные системы играют ключевую роль в обработке и анализе больших объемов данных, обеспечивая эффективное функционирование производства, предоставления, а также получения различных цифровых товаров и услуг. В условиях интеллектуализации и увеличения разнообразия функциональных требований, предъявляемых к информационным системам, возникает потребность в разработке новых подходов для их построения и реализации. В работе рассмотрены основные особенности применения предложенного агентно-модульного подхода, предполагающего построение гибких и масштабируемых информационно-вычислительных систем, способных функционировать в распределенной среде. Агентно-модульный подход представляет собой методологический подход к построению информационно-вычислительных систем, на основе интеграции методов системного анализа, а также агентного и модульного принципов организации систем. Цель работы – исследовать теоретические и практические аспекты использования предложенного агентно-модульного подхода, проанализировать его преимущества в сравнении с другими существующими подходами (объектным, компонентным, сервисным), а также представить пример построения информационной системы на основе его применения. Для достижения поставленной цели необходимо решить следующие задачи: – исследовать теоретические основы и выявить специфику практического применения предложенного агентно-модульного подхода; – выполнить анализ отличий и описать преимущества предложенного подхода при его сравнении с другими подходами; – представить пример построения информационно-вычислительной системы с использованием предложенного подхода. В процессе исследования были использованы следующие общенаучные методы: декомпозиция, анализ, синтез, сравнение, описание, формализация, структуризация, моделирование, проектирование, а также основные принципы модульного и агентного подходов. В качестве материалов исследования использовались компьютеры, инструментальные средства и программное обеспечение вычислительных систем. В результате исследования были рассмотрены основные теоретические и практические аспекты применения предложенного подхода. Агентно-модульный подход может быть использован для построения различных информационных систем на этапах, связанных с их моделированием и проектированием. Предложенный подход позволяет описывать структуру, функционирование, а также взаимодействие различных составных частей информационно-вычислительных систем.

Агентно-модульный подход; проектирование программных систем; разработка программных систем; информационные системы; вычислительные системы; обработка знаний; проектирование мультиагентных систем; системный анализ; информационное моделирование.

A.F. Zaytsev**FEATURES OF THE AGENT-MODULAR APPROACH APPLICATION IN THE DESIGN AND IMPLEMENTATION OF INFORMATION-COMPUTATIONAL SYSTEMS**

Information-computational systems play a key role in processing and analyzing large amounts of data, ensuring the efficient functioning of production, provision, as well as receipt of various digital goods and services. In the conditions of intellectualization and increasing diversity of functional requirements imposed on information systems, there is a need to develop new approaches for their construction and implementation. The paper considers the main features of the proposed agent-modular approach, which involves the construction of flexible and scalable information-computational systems capable of functioning in a distributed environment. The agent-modular approach is a methodological approach to the organization of information-computational systems based on the integration of system analysis methods, as well as agent and modular principle's of building various systems. The aim of the paper is to investigate theoretical and practical aspects of the application of the proposed agent-modular approach in building an information system, to analyze its advantages over other common approaches (object, component, service), and to present examples of its successful use. To achieve the set goal, it is necessary to solve the following tasks: – to research the theoretical foundations and identify the specific features of practical

application of the proposed agent-modular approach; – to analyze the differences and describe the advantages of the proposed approach in its comparison with other approaches; – to present an example of building an information-computational system using the proposed approach. The following general scientific methods were used in the research process: decomposition, analysis, synthesis, comparison, description, formalization, structurization, modeling, design, as well as the basic principles of system, modular and agent-based approaches. Computers, and computing software tools were used as research materials. As a result of the study, the main theoretical and practical aspects of the proposed approach were considered. The agent-modular approach can be used to build various information systems at the stages related to their modeling and design. The proposed approach allows describing the structure, functioning, and interaction of various components of information-computational systems.

Agent-modular approach; software systems design; software systems development; information systems; computational systems; knowledge processing; multiagent systems design; systems analysis; information modeling.

Введение. В настоящее время современные информационные технологии применяются во многих областях человеческой деятельности. Одним из наиболее перспективных ИТ-направлений является построение информационно-вычислительных и интеллектуальных систем обработки знаний. Подобные технические системы предназначены для автоматизации процессов анализа больших объемов данных и выдачи обоснованных результатов решений на базе множества знаний и закономерностей: принципов, правил, законов отображения, фактов, положений, убеждений и другой проверенной экспертной информации.

Для построения и реализации информационно-вычислительных систем используют различные существующие подходы (объектно-ориентированный, компонентно-ориентированный, сервисно-ориентированный, агентно-ориентированный), а также многие общеизвестные принципы (многоуровневости, иерархичности, модульности, разделения ответственности, открытой спецификации, openAPI и пр.) организации систем. Приведенные принципы и подходы являются основой для разработки более частных методик, приемов или паттернов проектирования систем [1]. В условиях интеллектуализации и увеличения разнообразия функциональных требований, предъявляемых к информационно-вычислительным системам, возникает потребность в разработке новых подходов к их построению и реализации. В текущей работе рассмотрены основные особенности применения предложенного агентно-модульного подхода, предполагающего построение гибких и масштабируемых информационно-вычислительных систем, способных функционировать в распределенной среде. Агентно-модульный подход представляет собой методологический подход к построению информационно-вычислительных систем, который основан на интеграции методов системного анализа, компьютерного логико-математического моделирования, а также агентного и модульного принципов организации систем. Первоначальная идея предложенного подхода была приведена в одной из ранних работ автора [2], но без каких-либо подробностей о его практическом применении.

Цель работы – исследовать теоретические и практические аспекты применения предложенного агентно-модульного подхода при построении информационно-вычислительных систем. Для достижения поставленной цели необходимо решить следующие задачи:

1. Исследовать теоретические основы и выявить специфику практического применения предложенного агентно-модульного подхода;
2. Описать основные отличия и преимущества предложенного подхода при его сравнении с другими подходами (объектным, компонентным, сервисным);
3. Представить пример построения информационно-вычислительной системы с использованием предложенного подхода.

Использование методологии системного анализа в контексте построения информационно-вычислительных систем. Практически все современные информационно-вычислительные системы строятся на базе обобщенного системного подхода и методологии системного анализа. Системный анализ представляет собой методологическую основу, которая может быть использована для решения задач: анализа, синтеза, проектирования, реализации, а также оценки эффективности (функционирования или качества) работы информационно-вычислительных систем.

Решение любой проблемы с использованием методологии системного анализа сопровождается определением и постановкой ряда отдельных задач, для решения которых требуется разработка и применение различных моделей, методов или методик. Процесс решения проблемы построения информационно-вычислительных систем с применением обобщенного системного подхода и методологии системного анализа можно отобразить в виде следующей схемы на рис. 1.

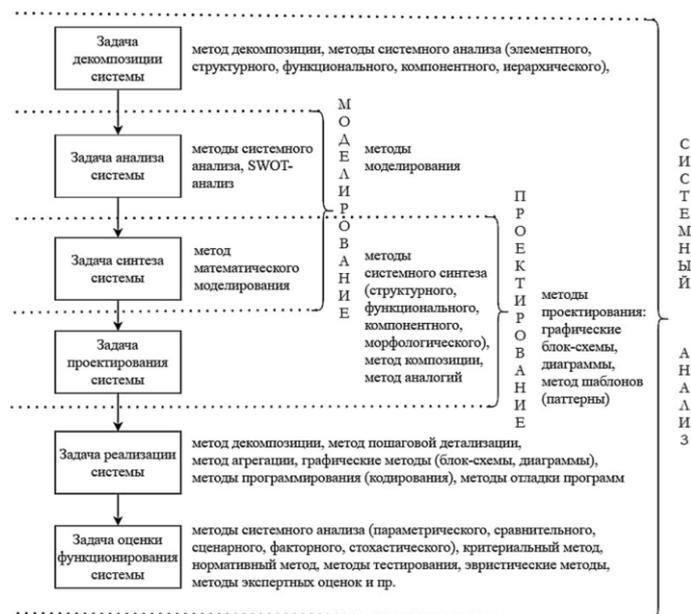


Рис. 1. Обобщенный системный подход к решению проблемы построения информационно-вычислительных систем с использованием методологии системного анализа

Множество общенаучных методов (и принципов) системного анализа служат основой для выработки и описания обобщенного системного подхода к решению существующей проблемы. Обобщенный системный подход отражает общий ход решения проблемы и представляет собой последовательность из согласованных этапов (шагов, стадий), как совокупности определённых, поставленных задач и применяемых методов для их решения.

Таким образом, применение различных методов системного анализа позволяет выполнять всесторонний анализ и синтез разрабатываемых информационно-вычислительных систем, выявлять оптимальные взаимосвязи между их составными частями и элементами, моделировать их структуру и архитектуру, а также на основе этого разрабатывать более эффективные алгоритмы анализа, управления и обработки информации.

Агентно-модульный подход и особенности его применения. На базе системного подхода был разработан и предложен агентно-модульный подход [2], который можно использовать для построения и реализации программных информационно-вычислительных систем. Предложенный методологический подход является частным случаем обобщенного системного подхода и использует только некоторые выбранные принципы и методы системного анализа. На рис. 2 приведена схема агентно-модульного подхода, в которую были добавлены корректировки и уточнения, относительно ранее представленного варианта [3].

Агентно-модульный подход – это методологический подход к построению и реализации информационно-вычислительных систем на основе обобщенной формальной модели, отражающей структурно-функциональную архитектуру системы в виде отдельных сущностей (агентов), которые после их алгоритмической реализации представляют собой автономные программные модули, способные взаимодействовать друг с другом.

Применение агентно-модульного подхода заключается в выполнении шести последовательных этапов (рис. 2), которые с использованием методологии системного анализа позволяют решить поставленную проблему и ключевые задачи: анализа, синтеза, проектирования и реализации системы. При этом одна часть этапов отражает процесс моделирования, а другая часть – процесс проектирования системы. Основной особенностью агентно-модульного подхода является совместное использование методов системного анализа [4], компьютерного логико-математического моделирования [5], а также агентно-ориентированного [6, 7] и модульного [8, 9] принципов организации программных информационно-вычислительных систем.



Рис. 2. Агентно-модульный подход к построению и реализации информационно-вычислительных систем

Основные отличия и преимущества агентно-модульного подхода. При сравнении предложенного подхода с уже существующими (объектно-ориентированным, компонентно-ориентированным и сервисно-ориентированным), можно выделить и описать ряд его принципиальных отличий.

Агентно-модульный подход использует принципы агентно-ориентированного подхода, который представляет собой обобщенную концепцию с введенным термином «агент» и его определением, как сущности с различными вариантами поведения, меняющимися в зависимости от состояния среды, в которой она находится [10]. В свою очередь, агентно-ориентированный подход является развитием объектно-ориентированного подхода, из чего следует то, что агенты взаимодействуют друг с другом через передачу сообщений. При использовании терминологии объектно-ориентированного подхода можно полагать, что агенты являются более сложными сущностями (составными объектами), которые состоят из множества взаимосвязанных простых объектов. Но в отличие от базового понятия «объект» агенты являются еще и автономными, интерактивными, а также обладающими своим внутренним состоянием и различными вариантами поведения. Основные отличия между понятиями «объект» и «агент» заключаются в следующем:

1) агенты ориентированы на открытие и предоставление информации (обмен данными), а объекты ориентированы на сокрытие информации (инкапсуляцию данных);

2) агенты ориентированы на прямое взаимодействие (сообщения с запросами и ответные реакции), как между внутренними частями одной системы, так и между внешними независимыми частями (агентами) других подобных систем. Объекты ориентированы только на взаимодействие между связанными внутренними частями (вложенность, полиморфизм и наследование объектов) одной системы;

3) агенты ориентированы на отдельный запуск и автономную работу (обработку данных) независимо от работы других частей (агентов) системы. Объекты не могут быть отдельно запущены и выполнять автономную работу независимо от других частей системы;

4) агенты не отслеживают состояния других агентов, не наследуются и не могут являться частями друг друга.

Кроме основных отличий, агенты также могут иметь такие дополнительные характеристики поведения, как: адаптивность, реактивность, проактивность, робастность, гибкость, инициативность, целенаправленность, коммуникативность, обучаемость и другие.

При использовании существующих формальных языков и парадигм (подходов) программирования [11], сущность агента может быть описана совокупностью различных элементов. Например, при использовании объектной парадигмы это могут быть взаимосвязанные классы с определенными свойствами и методами. При использовании функциональной парадигмы частями агента могут являться функции и функционалы с определёнными для них параметрами (переменными). При использовании структурной и процедурной парадигм сущность агента может быть описана структурами (совокупностью массивов переменных) и процедурами для их обработки. Таким образом, представление сущности агента в виде алгоритмической или программной реализации более правильно отождествлять с полнофункциональной программой, чем с отдельными составляющими: функциями, объектами, пакетами, библиотеками, компонентами или чем-то другим.

В предложенном агентно-модульном подходе с агентами отождествляются автономные программные модули – «агентные модули», содержащие заданное алгоритмическое описание сущности агента и конкретного механизма его взаимодействия с другими агентами, посредством передачи сообщений. На основании этого, в контексте построения программных информационно-вычислительных систем можно привести следующие определения агентного модуля и агента:

Агентный модуль – это автономный программный модуль, организованный в виде одного или нескольких связанных файлов, содержащих блоки с правильно оформленным кодом, записанным на некотором языке программирования. Блоки с кодом в файле агентного модуля алгоритмически отражают структуру и функционирование сущности агента в виде различных взаимосвязанных элементов: переменных, структур, классов, функций и т.п.

Агенты – это полностью завершённые, скомпилированные и автономно исполняемые агентные модули, то есть полнофункциональные работающие программы.

Агенты воспринимают и проецируют любые входящие сообщения на действия (функции), которые они могут самостоятельно выполнить, или на элементы состояния (параметры), которые могут повлиять на их дальнейшие действия. Агенты содержат только два вида реакции обратной связи: когда цель была ими достигнута, и когда цель не достигнута. В отличие от объектов, агенты могут принимать на себя обязательства по выполнению некоторых действий или, наоборот, отказываться от их выполнения. Агент не обязательно выполняет требования каких-либо других агентов или пользователей, он просто зависит от условий среды, своих целей и намерений других агентов. Функционирование информационно-вычислительной системы состоит из информирования множества агентов, принятия или отклонения ими различных предложений и требований, а также выполнения агентами определенных действий с последующими ответными реакциями (путем отправки и приема различных сообщений между независимыми и автономно исполняемыми модулями).

Отличия предложенного агентно-модульного подхода при его сравнении с компонентно-ориентированным и сервисно-ориентированным подходами построения информационно-вычислительных систем представлены на рис. 3.

В компонентно-ориентированном и сервисно-ориентированном подходах, кроме ключевых понятий «компонент» и «сервис», также встречается понятие модуля. Однако в данных существующих подходах понятие модуля отражает несколько иной смысл:

Модуль – это функциональная единица, представляющая собой совокупность взаимосвязанных файлов (пакетов и библиотек) с частями фрагментов кода на уровне написания программ. Модули реализуют функциональные возможности системы в виде за-

вершенных блоков кода с определением и реализацией различных отдельных методов, функций или объектов. В компонентном и сервисном подходах модули не являются полностью автономными исполняемыми (скомпилированными) программами.

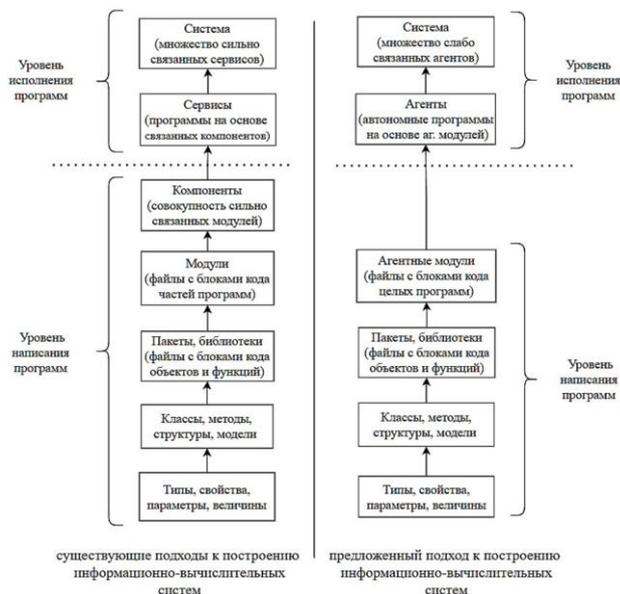


Рис. 3. Отличия подходов к построению информационно-вычислительных систем

Компонент – это конструктивная единица, являющаяся составной частью некоторой системы или её подсистемы. Компоненты предназначены для их повторного переиспользования и взаимосвязи с другими компонентами, содержащими определенные функциональные возможности модулей. Компоненты представляют собой совокупность из нескольких сильносвязанных модулей на уровне написания кода программ. В компонентах реализуется рекомендованный, заранее определенный и общий для всех интерфейс взаимодействия, соответствующий их единообразному поведению внутри системы.

Сервис – это подсистема, организованная в виде совокупности взаимосвязанных между собой независимых компонентов. Сервисы предназначены для выполнения конкретных бизнес-функций и предоставления пользователям различных услуг на уровне исполнения программ. Часто архитектура больших и сложных информационно-вычислительных систем представляется в виде множества сильносвязанных друг с другом сервисов (подсистем), как отдельно скомпилированных и работающих программ.

Основные отличия компонентов и сервисов от агентов заключаются в следующем:

- 1) агенты отражают только одну сущность предметной области и ее функциональные особенности поведения. Компоненты и сервисы соответствуют множеству связанных сущностей и понятий;
- 2) компоненты и сервисы сильно связаны между собой, а агенты связаны слабо и все функции взаимодействия должны быть отдельно определены в их интерфейсах;
- 3) интерфейс компонентов и сервисов сильно связан вместе с их реализацией. Реализация функционального поведения агентов и их интерфейсов – разграничены.

Построение системы с использованием предложенного подхода. В качестве результата применения предложенного подхода на рис. 4 представлена архитектура построенной информационно-вычислительной системы обработки знаний. Построенная система предназначена для поиска, формирования и вывода решений вместе с их численными результатами на основе знаний, полученных из постановок задач из области теоретической механики. Моделирование структурно-функциональной организации модулей системы, а также описание программной реализации базы данных и знаний было представлено в других предыдущих работах автора [12].

При использовании агентно-модульного подхода архитектура системы условно-логически разбивается на две части: внутреннюю и внешнюю. Внешняя часть представляется в виде отдельного модуля, взаимодействующего с окружающей средой, пользователями и внешними системами для приема/передачи сообщений через определенный интерфейс (графический и внутренний). Внутренняя часть представляется в виде множества автономно работающих агентных модулей (программ), взаимодействующих друг с другом через отдельно определенные (внутренние) интерфейсы: агент ввода/вывода, агент поиска решений, агент взаимодействия с базой данных и знаний, агент вычисления результатов решений и другие (рис. 4).

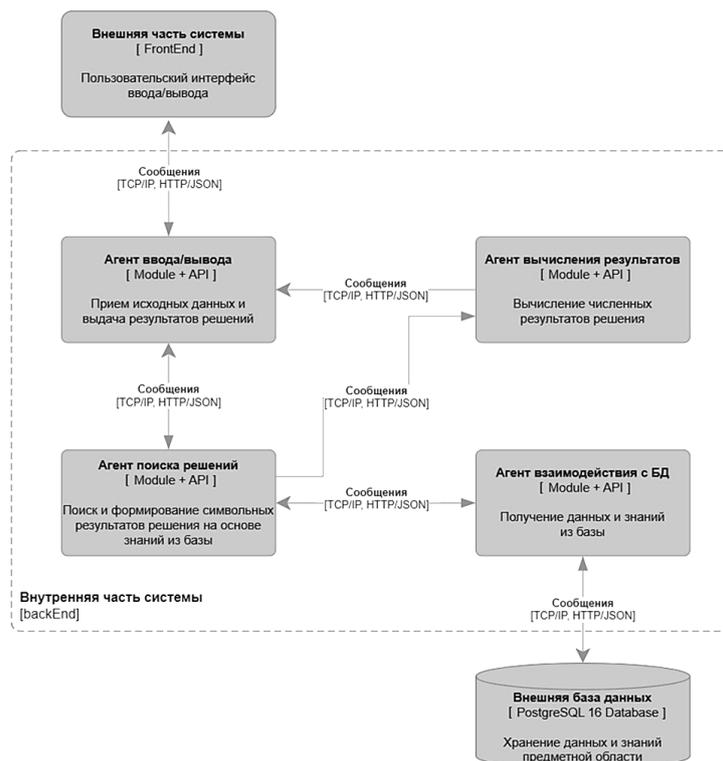


Рис. 4. Архитектура построенной информационно-вычислительной системы с применением агентно-модульного подхода

Каждый автономный агентный модуль для достижения своих основных целей выполняет множество определённых внутренних функций:

агент ввода/вывода – осуществляет прием исходных данных и выдачу полученных итоговых результатов решений задач;

агент поиска решений – выполняет поиск и формирование символьных результатов решений на основе данных и знаний из базы;

агент взаимодействия с базой – выполняет извлечение, предобработку и передачу необходимой информации (данных и знаний);

агент вычисления результатов решений – производит численные вычисления и передачу полученных итоговых результатов решений задач.

В состав системы можно добавлять и другие разнообразные агентные модули. Для связи модулей в каждом из них реализуется подмножество функций, которые образуют программный интерфейс их взаимодействия – API (Application Programming Interface). Таким образом, подмножество функций интерфейса в модуле могут вызывать подмножество других его внутренних функций, которые в свою очередь изменяют состояние и поведение сущности агента.

На различных алгоритмических языках программирования интерфейс (API) в модулях реализуется достаточно просто. Например, на языке Python (Питон) интерфейс взаимодействия в каждом модуле можно реализовать на базе следующих подключаемых пакетов: FastAPI, Django, Flask или других [13, 14]. В приведенном ниже листинге показан пример кода на языке Python [15, 16], отражающий реализацию функций интерфейса (call_input, call_output) агентного модуля с использованием подключаемого пакета Flask [17]:

```
from flask import Flask, jsonify, request, make_response
import requests
app = Flask()

@app.route('/call_input', methods=['GET', 'POST'])
def call_input():
    data = request.get_json()
    #выполнение множества других внутренних функций модуля...
    resp = requests.post(url='/call_search', json=data)
    if resp.status_code == 200:
        return jsonify({'success':True, 'act':'call_search',
'data':data})
    else:
        return jsonify({'success': False})

@app.route('/call_output', methods=['GET', 'POST'])
def call_output():
    #выполнение множества других внутренних функций модуля...
    return jsonify({'success':True, 'act':'call_output',
'data':data})
```

В процессе программной реализации агентных модулей можно выполнять их запуск и тестирование на одной локальной машине с указанием различных номеров для сетевых портов. Для реализации передачи информации можно использовать различные протоколы и технологии, например: TCP/IP, HTTPS, SSE, WebSocket, WebRTC, AsyncIO [18] и другие.

К преимуществам предложенного агнетно-модульного подхода можно также отнести следующее:

- 1) агентные модули могут иметь одноранговые (прямые и обратные) связи друг с другом, позволяющие строить мультиагентные системы [19] с симметричной архитектурой;
- 2) агентные модули могут быть реализованы независимо и на разных формальных языках программирования;
- 3) агентные модули организуются на основе множества «чистых» функций и всегда выполняют только свои внутренние базовые функции;
- 4) агентные модули имеют упрощенную организацию, их более удобно обновлять, заменять, а также дублировать и размещать в любое время;
- 5) агентные модули могут исполняться автономно в различных средах виртуализации и контейнеризации, а также на разных операционных системах и микропроцессорных архитектурах.

Каждый разработанный агентный модуль может быть запущен отдельно и работать автономно с использованием технологий контейнеризации, например, такой как Docker [20, 21]. Это дает возможность размещения агентных модулей на разных узлах (компьютерах) в распределенной среде и сети Интернет. При этом можно размещать несколько запасных копий агентных модулей, настраивать переадресацию и выполнять перенаправление данных между копиями в случае недоступности или отказа работы основных модулей.

Заключение. В результате исследования были рассмотрены основные теоретические и практические аспекты применения предложенного подхода. Агентно-модульный подход является эффективным средством для построения и реализации современных ин-

формационно-вычислительных систем. Разработанный подход позволяет описывать структуру, функционирование, а также взаимодействие различных составных частей информационно-вычислительных систем в виде агентных модулей. Подход может быть использован для построения различных информационных систем на этапах, связанных с их моделированием и проектированием.

Предложенный подход сочетает в себе преимущества принципов автономности и модульности, что позволяет строить гибкие и масштабируемые информационные системы. Применение предложенного подхода возможно в самых разных областях и направлениях, требующих построения программных информационных систем, начиная от выполнения локальных вычислений и заканчивая интеллектуальной обработкой глобально распределённых больших данных.

Дальнейшее совершенствование предложенного подхода связано с разработкой и реализацией методов координации, синхронизации и интеллектуализации действий агентных модулей (AI-агентов, ИИ-агентов) [22, 23], а также с унификацией и стандартизацией интерфейса для их взаимодействия. Это позволит ещё больше расширить область применения агентно-модульного подхода и сделать его основным подходом для построения сложных информационно-вычислительных и распределённых интеллектуальных систем.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Гамма Э., Хелм Р., Джонсон Р. и др. Паттерны объектно-ориентированного проектирования. Юбилейное издание. – СПб.: Питер, 2025. – 448 с.
2. Зайцев А.Ф. Агентно-модульный подход к проектированию и реализации программных систем на основе имитационного моделирования // System Analysis and Mathematical Modeling. – 2023. – Т. 5, № 3. – С. 338-349. – DOI: 10.17150/2713-1734.2023.5(3).338-349.
3. Зайцев А.Ф. Математическая модель структурно-функциональной организации программно-информационных вычислительных систем // Тенденции развития науки и образования. – 2024. – № 114-9. – С. 20-26. – DOI: 10.18411/trnio-10-2024-376.
4. Дреус Ю.Г. Основы системного анализа. – Вологда: Инфра-Инженерия, 2024. – 260 с.
5. Zaytsev A.F., Kravchenko V.A., Shirapov D.Sh. An approach to logical-mathematical computer modeling of linear and nonlinear dynamical systems // E3S Web of Conferences. – 2024. – Vol. 583. – P. 06014. – DOI: 10.1051/e3sconf/202458306014.
6. Принев М.А., Леденева Т.М., Гаршина В.В. Многоагентные системы: обзор современных подходов к моделированию и проектированию (Ч. 2) // Вестник Воронежского государственного университета. Серия: Системный анализ и информационные технологии. – 2024. – № 4. – С. 167-190. – DOI: 10.17308/sait/1995-5499/2024/4/167-190.
7. Голенков В.В., Шункевич Д.В. Агентно-ориентированные модели, методика и средства разработки совместимых решателей задач интеллектуальных систем // Программные продукты и системы. – 2020. – №3. – С. 404-412. – DOI: 10.15827/0236-235X.131.404-412.
8. Модульное программирование – РУВИКИ: научно-образовательный портал. – URL: https://ru.ruwiki.ru/wiki/Модульное_программирование (дата обращения: 01.06.2025).
9. Модульное программирование – Большая российская энциклопедия: научно-образовательный портал. – URL: <https://bigenc.ru/c/modul-noe-programmirovanie-ecb7b3/?v=3964503> (дата обращения: 01.06.2025).
10. Агентно-ориентированный подход – РУВИКИ: научно-образовательный портал. – URL: https://ru.ruwiki.ru/wiki/Агентно-ориентированный_подход (дата обращения: 10.06.2025).
11. Бабаев Б., Чарыева А., Чолуков Х. Парадигмы программирования: функциональное и объектно-ориентированное // Вестник науки. – 2024. – Т. 3, № 10 (79). – С. 877-880.
12. Зайцев А.Ф. Подход к преобразованию модели базы знаний предметной области на основе мультиграфа в реляционную модель базы данных // Системы управления, связи и безопасности. – 2024. – № 4. – С. 158-178. – DOI: 10.24412/2410-9916-2024-4-158-178.
13. Lubanovic B. FastAPI. Modern Python Web development. – Sebastopol: O'Reilly Media Inc., 2023. – 280 p.
14. Attardi J. Web API. Cookbook. – Sebastopol: O'Reilly Media Inc., 2024. – 286 p.
15. Manglik R. et al. Python notes for professionals. – Lucknow: EduGorilla publication, 2024. – 841 p.
16. Зайцев А.Ф. Преимущества и недостатки языка программирования Python при построении распределённых программных систем // Развитие современных технологий: теоретические и практические аспекты: Сб. статей V Международной научно-практической конференции, Петрозаводск, 14 октября 2024 года. – Петрозаводск: МЦНП «Новая Наука», 2024. – С. 73-77. – EDN N1H1ON.

17. *Tragura S.J.C.* Mastering Flask Web and API development. – Birmingham: Packt publishing, 2024. – 494 p.
18. *Fox D.* Python for asynchronous programming with AsyncIO. – Seattle: Amazon digital services LLC, 2024. – 126 p.
19. *Листопад С.В., Кириков И.А.* Архитектуры интеллектуальных агентов сплоченных гибридных интеллектуальных многоагентных систем // Системы и средства информатики. – 2022. – Т. 32, № 2. – С. 81-91. – DOI: 10.14357/08696527220208.
20. *Öggl B., Kofler M.* Docker: Practical guide for developers and DevOps teams. – Bonn: Rheinwerk Publishing Inc., 2023. – 491 p.
21. *Jones P.* Docker, Jenkins and Kubernetes: Mastering DevOps automation. – NY.: Walzone press, 2024. – 266 p.
22. *Тейлор М.* Эпоха ИИ-агентов. – Екатеринбург: Изд-во Ridero, 2025. – 84 с.
23. *Гохберг Л.М., Туровец Ю.В., Вишневецкий К.О.* Искусственный интеллект в России: технологии и рынки. – М.: НИУ ВШЭ, 2025. – 148 с. – DOI: 10.17323/978-5-7598-3019-1.

REFERENCES

1. *Gamma E., Khelm R., Dzhonson R. i dr.* Patterny ob"ektno-orientirovannogo proektirovaniya. Yubileynoe izdanie [Design Patterns: Elements of reusable object-oriented software]. Saint-Petersburg: Piter, 2025, 448 p.
2. *Zaytsev A.F.* Agentno-modul'nyy podkhod k proektirovaniyu i realizatsii programmykh sistem na osnove imitatsionnogo modelirovaniya [Agent-module approach to the design and implementation of software systems based on simulation modeling], *System Analysis and Mathematical Modeling*, 2023, Vol. 5, No. 3, pp. 338-349. DOI: 10.17150/2713-1734.2023.5(3).338-349.
3. *Zaytsev A.F.* Matematicheskaya model' strukturno-funktsional'noy organizatsii programmno-informatsionnykh vychislitel'nykh sistem [A mathematical model of the structural-functional organization of software-information computational systems], *Tendentsii razvitiya nauki i obrazovaniya* [Trends in Science and Education], 2024, No. 114-9, pp. 20-26. DOI: 10.18411/trnio-10-2024-376.
4. *Drevs Yu.G.* Osnovy sistemnogo analiza [Fundamentals of system analysis]. Vologda: Infra-Inzheneriya, 2024, 260 p.
5. *Zaytsev A.F., Kravchenko V.A., Shirapov D.Sh.* An approach to logical-mathematical computer modeling of linear and nonlinear dynamical systems, *E3S Web of Conferences*, 2024, Vol. 583, pp. 06014. DOI: 10.1051/e3sconf/202458306014.
6. *Prinev M.A., Ledeneva T.M., Garshina V.V.* Mnogoagentnye sistemy: obzor sovremennykh podkhodov k modelirovaniyu i proektirovaniyu (Ch. 2) [Multi-agent systems: a review of modern approaches to modeling and projecting (Part 2)], *Vestnik Voronezhskogo gosudarstvennogo universiteta. Seriya: Sistemnyy analiz i informatsionnye tekhnologii* [Proceedings of Voronezh State University. Series: Systems Analysis and Information Technologies], 2024, No. 4, pp. 167-190. DOI: 10.17308/sait/1995-5499/2024/4/167-190.
7. *Golenkov V.V., Shunkevich D.V.* Agentno-orientirovannye modeli, metodika i sredstva razrabotki sovместимых reshateley zadach intellektual'nykh sistem [Agent-based models, methods and tools for the development of compatible problem-solvers for intelligent systems], *Programmnyye produkty i sistemy* [Software & Systems], 2020, No. 3, pp. 404-412. DOI: 10.15827/0236-235X.131.404-412.
8. Modul'noe programmirovaniye – RUVIKI: nauchno-obrazovatel'nyy portal [Modular programming – RUVIKI: scientific and educational portal]. Available at: https://ru.ruwiki.ru/wiki/Modul'noe_programmirovaniye (accessed 01 June 2025).
9. Modul'noe programmirovaniye – Bol'shaya rossiyskaya entsiklopediya: nauchno-obrazovatel'nyy portal. Available at: <https://bigenc.ru/c/modul-noe-programmirovaniye-ecb7b3/?v=3964503> (accessed 01 June 2025).
10. Agentno-orientirovannyy podkhod – RUVIKI: nauchno-obrazovatel'nyy portal [Agent-oriented approach – RUVIKI: scientific and educational portal]. Available at: https://ru.ruwiki.ru/wiki/Agentno-orientirovannyy_podkhod (accessed 01 June 2025).
11. *Babaev B., Charyeva A., Cholukov Kh.* Paradigmy programmirovaniya: funktsional'noe i ob"ektno-orientirovannoe [Programming paradigms: functional and object-oriented], *Vestnik nauki* [Bulletin of Science], 2024, Vol. 3, No. 10 (79), pp. 877-880.
12. *Zaytsev A.F.* Podkhod k preobrazovaniyu modeli bazy znaniy predmetnoy oblasti na osnove mul'tigrafu v relyatsionnyuyu model' bazy dannykh [An approach to transforming a model of multigraph knowledge base of the subject area into a relational database model], *Sistemy upravleniya, svyazi i bezopasnosti* [Systems of Control, Communication and Security], 2024, No. 4, pp. 158-178. DOI: 10.24412/2410-9916-2024-4-158-178.

13. *Lubanovic B.* FastAPI. Modern Python Web development. Sebastopol: O'Reilly Media Inc., 2023, 280 p.
14. *Attardi J.* Web API. Cookbook. Sebastopol: O'Reilly Media Inc., 2024, 286 p.
15. *Manglik R. et al.* Python notes for professionals. Lucknow: EduGorilla publication, 2024, 841 p.
16. *Zaytsev A.F.* Preimushchestva i nedostatki yazyka programmirovaniya Python pri postroenii raspredelennykh programmnykh sistem [Advantages and disadvantages of Python programming language in developing distributed software systems], *Razvitie sovremennykh tekhnologiy: teoreticheskie i prakticheskie aspekty: Sb. statey V Mezhdunarodnoy nauchno-prakticheskoy konferentsii, Petrozavodsk, 14 oktyabrya 2024 goda* [Development of modern technologies: Theoretical and practical aspects, Proceedings of the V International Scientific and Practical Conference, Russia, Petrozavodsk, 14 October, 2024]. Petrozavodsk: MTSNP «Novaya Nauka», 2024, pp. 73-77. EDN NIHION.
17. *Tragura S.J.C.* Mastering Flask Web and API development. Birmingham: Packt publishing, 2024, 494 p.
18. *Fox D.* Python for asynchronous programming with AsyncIO. Seattle: Amazon digital services LLC, 2024, 126 p.
19. *Listopad S.V., Kirikov I.A.* Arkhitektury intellektual'nykh agentov splochnykh gibridnykh intellektual'nykh mnogoagentnykh sistem [Architectures of intelligent agents of cohesive hybrid intelligent multiagent systems], *Sistemy i sredstva informatiki* [Systems and Means of Informatics], 2022, Vol. 32, No. 2, pp. 81-91. DOI: 10.14357/08696527220208.
20. *Öggl B., Kofler M.* Docker: Practical guide for developers and DevOps teams. Bonn: Rheinwerk Publishing Inc., 2023, 491 p.
21. *Jones P.* Docker, Jenkins and Kubernetes: Mastering DevOps automation. NY.: Walzone press, 2024, 266 p.
22. *Taylor M.* Epokha II-agentov [The era of AI-Agents]. Ekaterinburg: Izd-vo Ridero, 2025, 84 p.
23. *Gokhberg L.M., Turovets Yu.V., Vishnevskiy K.O.* Iskusstvennyy intellekt v Rossii: tekhnologii i rynki [Artificial intelligence in Russia: technologies and markets]. Moscow: NIU VShE, 2025, 148 p. DOI: 10.17323/978-5-7598-3019-1.

Зайцев Анатолий Федорович – Восточно-Сибирский государственный университет технологий и управления; e-mail: lordsadler2010@mail.ru; г. Улан-Удэ, Россия; старший преподаватель кафедры «Вычислительные и радиоэлектронные системы».

Zaytsev Anatoly Fedorovich – East Siberia State University of Technology and Management; e-mail: lordsadler2010@mail.ru; Ulan-Ude, Russia; Senior Lecturer at the Department of «Computational and radioelectronic systems».

УДК 004.896

DOI 10.18522/2311-3103-2025-6-189-199

Д.Г. Веселова, Н.Е. Сергеев

СТОХАСТИЧЕСКАЯ МОДЕЛЬ КОГНИТИВНОГО РИСКА ТРЕЙДЕРОВ В УСЛОВИЯХ ВОЛАТИЛЬНОСТИ ДЕЦЕНТРАЛИЗОВАННЫХ РЫНКОВ

Настоящее исследование посвящено разработке стохастической модели когнитивного риска трейдеров как ключевого компонента интеллектуальной системы поддержки принятия решений (СППР) для децентрализованных криптовалютных рынков. Актуальность работы обусловлена спецификой DeFi-среды, характеризующейся высокой и нестационарной волатильностью, отсутствием централизованных механизмов стабилизации, асимметрией информации и значительным влиянием поведенческих факторов на принятие торговых решений. В данных условиях традиционные детерминированные и статические СППР демонстрируют ограниченную эффективность, поскольку не учитывают динамическое восприятие риска участниками рынка и связанные с ним когнитивные искажения. Целью исследования является формализация когнитивного риска трейдеров в виде стохастического процесса с эффектом памяти и интеграция соответствующей модели в архитектуру СППР, ориентированной на адаптивное управление риском. Для достижения поставленной цели разработано стохастическое дифференциальное уравнение, описывающее динамику когнитивного риска в зависимости от рыночной волатильности и текущих режимов рынка, а также вероятностное ядро переходов между рыночными состояниями, связывающее объективные рыночные характеристики с субъективным восприятием риска. Для оценки параметров модели предложен идентификационный каркас, основанный на алгоритме Expectation–Maximization в сочетании с фильтром частиц, что обеспечивает возможность работы с нели-