

Інтелектуальні системи управління транспортними системами.
Синергетичні системи екомобілів

УДК 656.051

DOI: 10.30977/VEIT.2018.14.0.35

**ИСПОЛЬЗОВАНИЕ МУЛЬТИАГЕНТНЫХ СИСТЕМ В
ТРАНСПОРТНОЙ ЛОГИСТИКЕ**

Пронин С. В.¹

¹Харьковский национальный автомобильно-дорожный университет

***Аннотация:** Рассматривается подход к созданию систем поддержки принятия решений на основе мультиагентного подхода для решения логистических задач на автомобильном транспорте. В статье проанализированы инструменты для разработки агентных систем. Сформирована общий принцип работы многоагентной системы для организации перевозки грузов на автомобильном транспорте.*

***Ключевые слова:** искусственный агент; перевозки; транспортная логистика системы поддержки принятия решений.*

Введение

Быстрое развитие и распространение новых технологий предполагает необходимость в новом качественном программном обеспечении. в том числе автоматизация бизнес-процессов в логистике.

На рынке автоматизации логистики в 2017-2018 годах наблюдается все больше и больше запросов на ERP, TMS, FMS и WMS системы, вырос спрос на роботизацию, технологии Data Science, дополненную реальность и IoT

Для повышения эффективности бизнеса в сфере транспортной логистики необходимо детальное изучение материальных (товары и грузы) и информационных потоков (данные по заказам и поставкам, финансовые данные и т.п.).

По подсчетам аналитиков логистика может составлять до 40% себестоимости продукции. Методы минимизации затрат на логистику различны.

Использование информационных технологий в логистике позволят решить ряд таких как:

- экономия времени;
- максимальное упрощение работы;
- визуализация результатов работы и выявление слабых мест в схеме;

- сокращение издержек.

Основная задача автоматизации в сфере транспортной логистики – расчет оптимальных маршрутов с учетом множества неравнозначных критериев. Экономия достигается не только благодаря сокращению издержек на транспорт, но и за счет оптимизации графика работы перевозчиков, позволяющего избегать простоев и недогруза.

В данной статье рассматривается подход к созданию систем с помощью многоагентного подхода.

Цель исследования

Для решения задачи в данной статье предложено использовать технологию многоагентных систем. Данный подход основан на использовании специальных автономных индивидуумов (агентов), объединенных в систему. Агент здесь представляет из себя программу, которая в автономном режиме способна осуществлять самостоятельные действия в соответствии с заданной целью.

Целью исследования является определение архитектуры системы, анализ современных программных сред для разработки многоагентных систем, описание поведения искусственных агентов.

Общий принцип работы агентной системы для решения задач организации перевозки грузов

Сценарий работы системы поддержки принятия решения, рассматривающийся в данной статье будет включать в себя несколько типов искусственных агентов таких как агенты-перевозчики и агенты-пользователи транспортных услуг.

Каждый агент-пользователь транспортных услуг получает заносит в общее информационное пространство данные о географии перевозки, составе груза и финансовых условиях, и периодически запрашивает предложение (услуги на перевозку) у всех известных ему агентов-перевозчиков. Когда предложение получено, агент-пользователь транспортных услуг принимает его и выдает заказ на перевозку. Если несколько агентов-перевозчиков предоставляют предложения агенту-пользователю транспортных услуг, он принимает лучшее из них.

Каждый агентов-перевозчиков непрерывно прослушивает запросы от агентов-пользователей транспортных услуг. При получении запроса на перевозку агенты-перевозчики проверяют свои заказы и, на основе общего списка заказов могут формировать оптимальный маршрут для перевозки. Если условия по выполнению конкретного заказа не отвечают требованиям, то заказ отклоняется. После выполнения заказа на перевозку он удаляется из общего списка.

Анализ литературы

Для решения поставленной задачи подходит четыре типа программных агентов [1-2]:

- агенты-покупатели или торговые боты;
- пользовательские или персональные агенты;
- агенты по мониторингу и наблюдению;
- агенты по добыче и анализу данных.

Агенты-покупатели в основном просматривают сетевые ресурсы с целью получения информации о товарах и услугах базируясь на семантической информации, представляемой в сети [3] и программных интерфейсах (API) откуда они получают уведомления.

Для информационных задач так называемых чатботов (Telegram, Twitter, Facebook Messenger, etc. [4])

Пользовательские или персональные агенты представляют из себя интеллектуальные агенты, которые действуют от имени пользователя. Типичные примеры – отправка данных, получение и автоматическая обработка данных.

Агенты по мониторингу и наблюдению используются для наблюдения за объектами и передачи информации на оборудование. Здесь термин агент употребляется, например, в SNMP мониторинге [5]. Применительно к рассматриваемым в статье задачам агенты отслеживать спрос и предложение на перевозку грузов.

Агенты по добыче и анализу данных представляют из себя, программы (сервисы), которые обрабатывают данные из нескольких источников (а именно это и характерно для агентов) чаще всего называют мэшапами. Все современные телекоммуникационные сервисы (Telecom 2.0), а также сервисы Интернета Вещей представляют из себя именно мэшапы [6].

Инструменты для разработки искусственных агентов

Формы распределения и взаимодействия агентов исследуется различными группами ученых, работающими над стандартизацией взаимодействия мультиагентных систем. Одну из основных групп представляют собой FIPA (Foundation for Intelligent Physical Agents)[7], OMG (Object Management Group), KAoS (Knowledge-able Agent- oriented System) и другие. (рис. 1).

The Foundation for Intelligent Physical Agents (FIPA) - это сообщество разработчиков, целью которых является стандартизация агентных технологий. Эта организация создала ряд спецификаций для непосредственного использования в мультиагентных системах. Самыми главными среди этих спецификаций являются Управление Агентами (Agent Management) и Язык Коммуникации Агентов (Agent Communication Language).

Управление агентами предусматривает использование нормативного фреймворка, внутри которого FIPA-агенты существуют и взаимодействуют. Он определяет логические

модели для создания, регистрации, определения местонахождения, коммуникации, миграции и удаления агентов.

Сущности, определяемые в этой модели – это не что иное, как набор каких-то возможностей (служб), которые вовсе не связаны с физической конфигурацией. Вдобавок, особенности реализации каждой конкретной агентной платформы и агентов являются строго выбором разработчиков, занимающихся построением конечной мультиагентной системы) [1].

Модель управления агентами состоит из следующих логических компонент, каждая из которых представляет набор определенных возможностей (рис. 2):

Агент – это некий вычислительный процесс, который наделен автономностью и коммуникативной функциональностью в рамках приложения.

Агенты обмениваются информацией с помощью Языка Коммуникации Агентов (Agent Communication Language). Агенты – это главные действующие сущности агентной платформы, которые представляют собой комбинацию одной или множества различных сервисных возможностей и операций, описанных в их характеристиках, внутри объединенной и интегрированной исполняемой модели. Агент должен быть привязан к конкретному пользователю, и выражать хотя бы одно понятие для своей идентификации которое называется Идентификатор Агента (Agent Identifier (AID)), который однозначно определяет агента в Множестве Агентов (Agent Universe). Агент может быть зарегистрирован как набор транспортных адресов, с которыми он может контактировать.

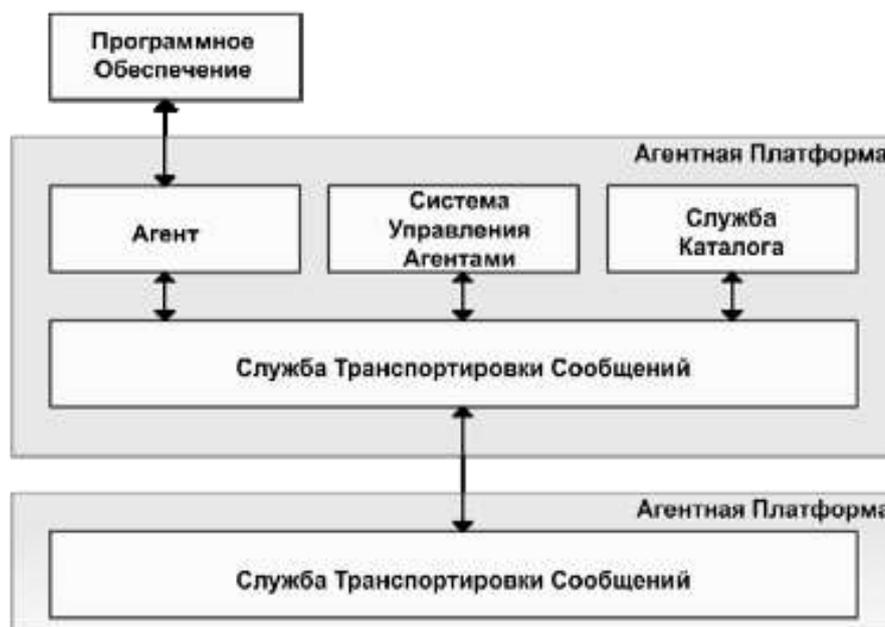


Рис. 1. Модель управления агентами FIPA

Служба Каталога (Directory Facilitator (DF)) – это опциональный компонент Агентной Платформы, но если он присутствует, то он реализуется как специальная служба-куратор. Служба Каталога представляет собой «желтые страницы» возможностей других агентов. Агенты могут регистрировать свои возможности с помощью службы каталога или запрашивать список доступных для выполнения чего-либо агентов. Внутри Агент-

ной платформы могут существовать множественные службы каталога, и они могут объединяться в федерации.

Система Управления Агентами (Agent Management System (AMS)) – это компонент агентного приложения. Система управления агентами проводит непосредственный контроль за существованием и использованием Агентной платформы. Система Управления Агентами обслуживает каталог идентифика-

торов агентов, которые содержат транспортные адреса для зарегистрированных в Агентной Платформе агентов. Она представляет собой «белые страницы» возможностей для других агентов. Каждый агент регистрируется в Системе Управления Агентами для получения действительного идентификатора.

Служба Транспортировки Сообщений (Message Transport Service (MTS)) - это стандартный коммуникационный метод коммуникации между агентами различных Агентных Платформ.

Агентная Платформа (Agent Platform (AP)) представляет физическую инфраструктуру, в которой могут быть развернуты агенты. Агентная Платформа состоит из вычислительной техники, операционной системы, программного обеспечения поддержки агентов, компонентов управления FIPA-агентами (Служба Каталога, Система Управления Агентами и Служба Передачи Сообщений) и самих агентов.

Внутренняя структура Агентной Платформы зависит от системных разработчиков и не является предметом стандартизации FIPA. Агентные Платформы и агенты этой платформы (как созданные, так и появившиеся путем миграции) могут использовать любые собственные методы взаимодействия.

FIPA рассматривается только с точки зрения коммуникаций между агентами как внутри, так и извне для агентной платформы. Агенты свободно могут обмениваться сообщениями любыми поддерживаемыми способами.

Агенты могут получать доступ к программному обеспечению, к примеру, для добавления новых возможностей, запроса новых коммуникационных протоколов, запроса новых протоколов и алгоритмов шифрования, запроса новых протоколов согласования, и т.д.

Основная идея платформы – выделение общих уровней, отделение коммуникаций (транспорта) от бизнес-логики.

Как конкретный пример реализации этой спецификации можно назвать JADE (Java Agent Development) – Open Source пакет для создания программных агентов [8], который реализован на языке Java. Технически представляет из себя программный сервер, на котором исполняются агенты. Помимо среды для запуска агентов, в JADE входят библиотеки (в Java – пакеты) для разработки агентов и графические инструменты для администрирования и мониторинга.

Общая структура представлена на рис. 2.

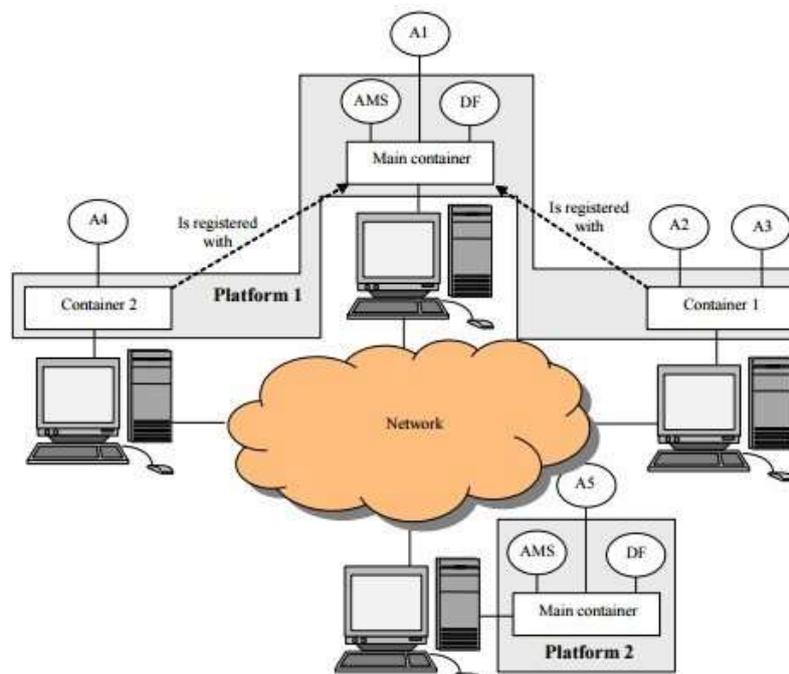


Рис. 2. Контейнеры JADE

Основными элементами JADE являются:

- контейнер. Это запущенная копия JADE, которая может содержать несколько агентов;
- платформа. Это совокупность активных контейнеров. Среди активных контейнеров выбирается один главный, и он и содержит информацию обо всех остальных контейнерах.

Каждый агент в системе характеризуется своим уникальным именем. Главный контейнер (main container) в JADE запускает два специальных агента. Можно назвать их служебными агентами: - AMS (Agent Management System). Этот агент обеспечивает сервис управления другими агентами. Например, с его помощью можно создать (запустить) или остановить (удалить) агента;

DF (Directory Facilitator). Этот агент подде-

рживает каталог агентов (“Желтые страницы”). Здесь агенты смогут искать других агентов, которые им понадобятся для достижения целей.

Наличие каталога агентов и единственного главного контейнера платформы позволяет агентам общаться (обмениваться информацией) между собой. При этом такое взаимодействие возможно внутри контейнера, между контейнерами одной платформы (между агентами разных контейнеров на одной платформе) и между платформами (между агентами в контейнерах разных платформ). Обмен сообщениями осуществляется асинхронно, с использованием очередей (рис. 3).

Поведение конкретного агента в системе – это отдельная задача. В JADE эти процессы отличаются от базовой модели процессов (Thread) в Java.

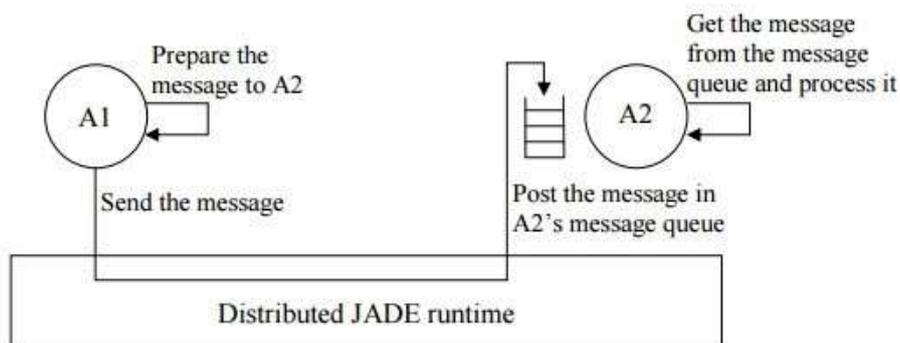


Рис. 3. Обмен сообщениями в JADE

Программист должен определять, когда заканчивается одно исполнение и начинается следующее. Это позволяет сохранять жесткое соответствие: один агент – один Java процесс. Естественно, что это работает быстрее, чем переключение между процессами, а также позволяет избежать проблем с синхронизацией

(например, если бы мы имели несколько процессов для одного агента, которые бы работали, естественно, с одним и тем же ресурсом) [9].

На рис. 4 показана организация каталога агентов.

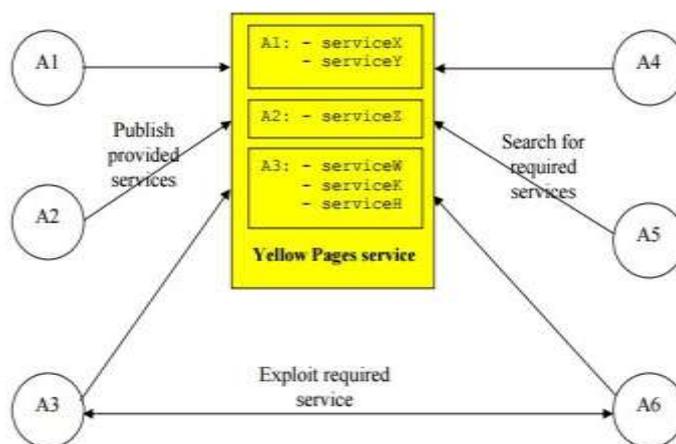


Рис. 4. Каталог в JADE

Планирование и исполнение поведений искусственным агентом

Работа агента должна строиться, как правило, в рамках заложенной в него поведенческой модели. Такая модель представляет собой задание, которое агент может выполнить, и фреймверке JADE реализуется как объект класса, наследуемый от *jade.core.behaviours.Behaviour*. Для того, чтобы агент выполнял задание, описанное в объекте поведения необходимо добавить поведение агента методом *addBehaviour()* класса агента. Поведение может быть добавлено в любой момент: когда запускается агент (метод *Setup()*) или в рамках другой модели поведения.

В каждый поведенческий класс, наследуемый от класса *Behaviour*, включаются методы *action()* и *done()*. Первый – позволяет описать действия, выполняемые агентом. Второй – возвращает признак завершенности действий, составляющих поведение агента.

При выполнении агентом задачи объект класса *Behavior()* удаляется из списка всех подобных объектов. В общем случае у агента может существовать несколько линий поведения, которым будут соответствовать разные поведенческие объекты, при этом агент может демонстрировать разные линии поведения параллельно. В такой ситуации различным линиям поведения не устанавливаются различные приоритеты, как для *java*-потоков а выполнение происходит кооперативно. Метод *action()* всегда выполняется от начала до конца и никогда не завершает свою работу самостоятельно. Если в текущий момент список поведенческих объектов пуст, то агент выгружается из памяти («засыпает») и не занимает ресурсы. Агент «пробуждается», если в списке появляется новый объект.

Хотя это требует дополнительных усилий при написании программного кода, такой подход имеет несколько преимуществ.

- позволяет работать с одним потоком *Java* для агента (что является весьма важным, особенно в условиях с ограниченными ресурсами, такими какими обладают сотовые телефоны).

- обеспечивает лучшее исполнение, так как переключения между поведением происходят гораздо быстрее, чем

переключения между потоками *Java*.

- устраняет все вопросы синхронизации между параллельными поведением для доступа к одним и тем же ресурсам (это ускоряет производительность тоже), поскольку все поведенья выполнялись одним и тем же потоком *Java*.

- когда поведение переключается, статус агента не содержит какого-либо стека информации, и поэтому можно получить "снимок" от него. Это позволяет осуществлять важные дополнительные свойства, например сохранение статуса агента в постоянном хранилище для последующего возобновления (постоянный агент), или передачи его другому контейнеру для удаленного исполнения (мобильный агент).

- когда нет доступных поведений для исполнения агентом, поток "засыпает", чтобы не занимать процессорное время. Он "просыпается", как только снова появляется поведение доступное для исполнения

Выводы

В статье проанализирован подход к созданию системы обмена информацией между участниками транспортного процесса, в сфере грузовых перевозок с использованием мультиагентного подхода.

Проанализированы инструменты для построения агентных систем.

Рассмотрены вопросы связанные с особенностью организации взаимодействия между интеллектуальными агентами.

Литература

1. Намиот Д. Е., Сухомлин В. А., Шаргалин С. П. Программные агенты в ERP системах. *Journal of Open Information Technologies* ISSN: 2307-8162 vol. 4, no. 6, 2016.
2. Cummings M., Haag S., McCubbrey D. *Management information systems for the information age.* – 2003.
3. Ristoski P., Paulheim H., Semantic Web in data mining and knowledge discovery: A comprehensive survey. *Web Semantics: Science, Services and Agents on the World Wide Web.* – 2016.
4. Namiot D. Twitter as a transport layer platform. *Artificial Intelligence and Natural Language and Information Extraction, Social*

- Media and Web Search FRUCT Conference (AINL-ISMW FRUCT)*, 2015. – IEEE, 2015. – С. 46-51.
5. Stallings W. SNMP, SNMPv2, SNMPv3, and RMON 1 and 2. *Addison-Wesley Longman Publishing Co., Inc.*, 1998.
 6. Namiot D., Sneys-Snepp M. On software standards for smart cities: API or DPI. *ITU Kaleidoscope Academic Conference: Living in a converged world-Impossible without standards*, Proceedings of the 2014. – IEEE, 2014. – С. 169-174.
 7. FIPA URL: <http://www.fipa.org> (дата звернення 23.12.2018).
 8. JADE URL: <http://jade.tilab.com/> (дата звернення 23.12.2018).
 9. Fabio Bellifemine, Giovanni Caire, Tiziana Trucco (TILAB, formerly CSELT), Giovanni Rimassa (University of Parma) JADE programmer's guide. Retrieved from: http://jade.tilab.com/doc/programmers_guide.pdf (accessed: 23.12.2018).

References

1. Namiot D.E., Sukhomlin V.A., Sharghalin S.P. (2016). Software Programnie agenti v ERP systemach [Software Agents in ERP Systems]. *International Journal of Open Information Technologies* ISSN: 2307-8162 vol. 4, no. 6th. [in Russian].
2. Cummings M., Haag S., McCubbrey D. (2003). Management information systems for the information age.
3. Ristoski P., Paulheim H. (2016). Semantic Web in data mining and knowledge discovery: A comprehensive survey. *Web Semantics: Science, Services and Agents on the World Wide Web*.
4. Namiot D. (2015). Twitter as a transport layer platform. *Artificial Intelligence and Natural Language and Information Extraction, Social Media and Web Search FRUCT Conference (AINL-ISMW FRUCT)*, 2015. – IEEE.
5. Stallings W. (1998). SNMP, SNMPv2, SNMPv3, and RMON 1 and 2. – Addison-Wesley Longman Publishing Co., Inc.
6. Namiot D., Sneys-Snepp M. (2014). On software standards for smart cities: API or DPI. *ITU Kaleidoscope Academic Conference: Living in a converged world-Impossible without standards?*, Proceedings of the 2014. – IEEE, 2014. 169-174.
7. FIPA Retrieved from: <http://www.fipa.org> (accessed: 23.12.2018).

8. JADE Retrieved from: <http://jade.tilab.com/> (accessed: 23.12.2018).
9. Fabio Bellifemine, Giovanni Caire, Tiziana Trucco (TILAB, formerly CSELT), Giovanni Rimassa (University of Parma) JADE programmer's guide. Retrieved from: http://jade.tilab.com/doc/programmers_guide.pdf (accessed: 23.12.2018).

Пронин Сергей Викторович¹, к.т.н., доц., кафедра компьютерных технологий и мехатроники, тел: (057) 707-37-43, email: sergiy9977@ukr.net,

¹Харьковский национальный автомобильно-дорожный университет, Украина, Харьков, 61002, ул. Ярослава Мудрого, 25.

Using multiagent systems in transport logistics

Abstract: *The rapid development and diffusion of new technologies implies the need for new high-quality software. including automation of business processes in logistics. In the market of automation of logistics in 2017-2018, there are more and more requests for ERP, TMS, FMS and WMS systems, the demand for robotization, Data Science technology, augmented reality and IoT has increased. To improve the efficiency of business in the field of transport logistics, a detailed study of material (goods and goods) and information flows (data on orders and deliveries, financial data, etc.) is necessary. According to analysts, logistics can be up to 40% of the cost of production. Methods to minimize the cost of logistics are different. The use of information technology in logistics will solve a number of such as: saving time; maximum simplification of work; visualization of work results and identification of weak points in the scheme; reduction of costs. The main task of automation in the field of transport logistics is the calculation of optimal routes, taking into account the set of unequal criteria. The savings are achieved not only due to the reduction of transportation costs, but also due to the optimization of the carrier's work schedule, which allows avoiding downtime and underloading. The scenario of the decision support system considered in this article will include several types of artificial agents such as transport agents and transport service user agents. Each agent-user of transport services receives data on the geography of transportation, composition of the cargo and financial conditions into the general information space, and*

periodically requests a proposal (services for transportation) from all agents-carriers known to it. When the offer is received, the transport service user agent accepts it and issues a freight order. If several transport agents provide transport service offers to the user agent, it accepts the best of them. Each carrier agent continuously listens to requests from transport service user agents. Upon receipt of a request for transportation, carrier agents check their orders and, based on a general list of orders, can form the best route for transportation. If the conditions for the implementation of a specific order do not meet the requirements, then the order is rejected. After completing the order for transportation, it is removed from the general list. **Goal:** Determination of the system architecture, analysis of modern software environments for the development of multi-agent systems, description of the behavior of artificial agents. **Methodology:** To solve the problem in this article it is proposed to use the technology of multi-agent systems. This approach is based on the use of special autonomous individuals (agents), united in the system. The agent here is a program that is offline capable of performing independent actions in accordance with a given goal. **Result:** The article analyzes the approach to creating a system for the exchange of information between participants in the transport process in the field of freight traffic using a multi-agent approach. Analyzed tools for building agent systems. Issues related to the peculiarity of the organization of interaction between intelligent agents are considered. **Originality:** the use of agent-based approach to solve the problem of information interaction of participants in the transport process. **Practical value:** the possibility of developing a software product to solve problems of the organization of transportation of goods.

Keywords: artificial agent; transportation; transport logistics decision support systems.

Pronin Sergey¹, Ph.D., Assoc. Prof., Department of Computer Technologies and Mechanics, tel: (057) 707-37-43, email: sergiy9977@ukr.net,
¹Kharkov National Automobile and Highway University, 25, Yaroslav Mudry street, Kharkiv, 61002, Ukraine.

Використання мультиагентних систем в

транспортній логістиці

Анотація: Швидкий розвиток і поширення нових технологій передбачає необхідність в новому якісному програмному забезпеченні. в тому числі автоматизація бізнес-процесів в логістиці. На ринку автоматизації логістики в 2017-2018 роках спостерігається все більше і більше запитів на ERP, TMS, FMS і WMS системи, виріс попит на роботизацію, технології Data Science, доповнену реальність і IoT. Для підвищення ефективності бізнесу в сфері транспортної логістики необхідно детальне вивчення матеріальних (товари та вантажі) і інформаційних потоків (дані по замовленнях і постачання, фінансові дані і т.п.). За підрахунками аналітиків логістика може становити до 40% собівартості продукції. Методи мінімізації витрат на логістику різні. Використання інформаційних технологій в логістиці дозволять вирішити ряд таких як: економія час; максимальне спрощення роботи; візуалізація результатів роботи і виявлення слабких місць в схемі; скорочення витрат. Основне завдання автоматизації в сфері транспортної логістики – розрахунок оптимальних маршрутів з урахуванням безлічі нерівнозначних критеріїв. Економія досягається не тільки завдяки скороченню витрат на транспорт, але і за рахунок оптимізації графіка роботи перевізників, що дозволяє уникати простой і недовантаження. Для вирішення завдання в даній статті запропоновано використовувати технологію багатоагентних систем. Даний підхід заснований на використанні спеціальних автономних індивідуумів (агентів), об'єднаних в систему. Агент тут представляє з себе програму, яка в автономному режимі здатна здійснювати самостійні дії відповідно до заданої метою. Метою дослідження є визначення архітектури системи, аналіз сучасних програмних середовищ для розробки багатоагентних систем, опис поведінки штучних агентів. Сценарій роботи системи підтримки прийняття рішення, розглядається в даній статті буде включати в себе кілька типів штучних агентів таких як агенти-перевізники і агенти-користувачі транспортних послуг. Кожен агент-користувач транспортних послуг отримує

заносить в загальний інформаційний простір дані про географію перевезення, складі вантажу і фінансових умовах, і періодично запитує пропозицію (послуги на перевезення) у всіх відомих йому агентів-перевізників. Коли пропозиція отримано, агент-користувач транспортних послуг приймає його і видає замовлення на перевезення. Якщо кілька агентів-перевізників надають пропозиції агенту-користувачеві транспортних послуг, він приймає найкраще з них. Кожен агентів-перевізників безперервно прослуховує запити від агентів-користувачів транспортних послуг. При отриманні запиту на перевезення агенти-перевізники перевіряють свої замовлення і, на основі загального списку замовлень можуть формувати оптимальний

маршрут для перевезення. Якщо умови щодо виконання конкретного замовлення не відповідають вимогам, то замовлення відхиляється. Після виконання замовлення на перевозкуон видається із загального списку. **Ключові слова:** штучний агент; перевезення; транспортна логістика системи підтримки прийняття рішень.

Пронін Сергій Вікторович¹, к.т.н., доц., кафедра комп'ютерних технологій і мехатроніки, тел: (057) 707-37-43, email: sergiy9977@ukr.net,

¹Харківський національний автомобільно-дорожній університет, Україна, Харків, 61002, вул. Ярослава Мудрого, 25.