

Формування математичної моделі функціонування транспортно-пересадочних вузлів міського пасажирського транспорту / В. О. Вдовиченко, Г. О. Самчук // Вісник НТУ «ХПІ». Серія: Механіко-технологічні системи та комплекси. – Харків : НТУ «ХПІ», 2016. – No 17(1189). – С.56–61. – Бібліогр.: 11 назв. – ISSN 2079-5459.

Формирование математической модели функционирования транспортно-пересадочных узлов городского пассажирского транспорта / В. А. Вдовиченко, А. А. Самчук // Вісник НТУ «ХПІ». Серія: Механіко-технологічні системи та комплекси. – Харків : НТУ «ХПІ», 2016. – No 17(1189). – С.56–61. – Бібліогр.: 11 назв. – ISSN 2079-5459.

Development of a mathematical model of public transport interchanges functioning/ V.O. Vdovychenko, G. O. Samchuk // Bulletin of NTU “KhPI”. Series: Mechanical-technological systems and complexes. – Kharkov: NTU “KhPI”, 2016. – No 17 (1189). – P.56–61. – Bibliogr.: 11. – ISSN 2079-5459.

Відомості про авторів / Сведения об авторах / About the Authors

Вдовиченко Володимир Олексійович – кандидат технічних наук, доцент, Харківський національний автомобільно-дорожній університет, доцент кафедри транспортних технологій; вул. Ярослава Мудрого, 25, м. Харків, Україна, 61002; тел.: (057) 707-37-20; e-mail: vval2301@gmail.com.

Вдовиченко Владимир Алексеевич – кандидат технических наук, доцент, Харьковский национальный автомобильно-дорожный университет, доцент кафедры транспортных технологий; ул. Ярослава Мудрого, 25, г. Харьков, Украина, 61002; тел.: (057) 707-37-20; e-mail: vval2301@gmail.com.

Vdovychenko Volodymyr – Candidate of Technical Sciences (Ph. D.), Docent, Kharkiv National Automobile and Highway University, Associate Professor at the Department of Transport Technologies; st. Yaroslav Mydrogo, 25, Kharkiv, Ukraine, 61002; tel.: (057) 707-37-20; e-mail: vval2301@gmail.com.

Самчук Ганна Олександрівна – Харківський національний автомобільно-дорожній університет, аспірант кафедри транспортних технологій; вул. Ярослава Мудрого, 25, м. Харків, Україна, 61002; тел.: (057) 707-37-20; e-mail: ganna.samchuk@gmail.com.

Самчук Анна Александровна – Харьковский национальный автомобильно-дорожный университет, аспирант кафедры транспортных технологий; ул. Ярослава Мудрого, 25, г. Харьков, Украина, 61002; тел.: (057) 707-37-20; e-mail: ganna.samchuk@gmail.com.

Samchuk Ganna – Kharkiv National Automobile and Highway University, Post Graduate at the Department of Transport Technologies; st. Yaroslav Mydrogo, 25, Kharkiv, Ukraine, 61002

УДК 004.94; 519.876

Р. Ю. ЛОПАТКИН, С. А. ПЕТРОВ, С. Н. ИГНАТЕНКО, В. А. ИВАЩЕНКО

ПЕРСПЕКТИВЫ ПРИМЕНЕНИЯ ИМИТАЦИОННОГО МОДЕЛИРОВАНИЯ В ЗАДАЧАХ АВТОМАТИЗАЦИИ И УПРАВЛЕНИЯ ТЕХНОЛОГИЧЕСКИМИ СИСТЕМАМИ

В работе рассматриваются фундаментальные аспекты применения имитационного моделирования в задачах автоматизации и управления техническими системами и комплексами, рассмотрены и проанализированы стандарты и существующие протоколы построения подобных систем. Исследован вопрос развития систем имитационного моделирования в разрезе развития науки в области синтеза информационных систем, структурного проектирования, рассматривается связь с системами искусственного интеллекта. Определены принципиальные характеристики систем имитационного моделирования а также обосновывается перспективность применения распределенных, масштабируемых систем на базе мульти-агентного подхода.

Ключевые слова: имитационное моделирование, мульти-агентные системы, моделирование управления, агентное моделирование, распределенные агентные системы.

В роботі розглядаються фундаментальні аспекти застосування імітаційного моделювання в задачах автоматизації і керування технічними системами і комплексами. Розглянуто та проаналізовано стандарти та протоколи побудови подібних систем. Досліджено питання розвитку систем імітаційного моделювання в розрізі розвитку науки в галузі синтезу інформаційних систем, структурного проектування та зв'язку з системами штучного інтелекту. Визначені принципові характеристики систем імітаційного моделювання а також обґрунтовується перспектива застосування розподілених систем на базі мульти-агентного підходу що масштабується.

Ключові слова: імітаційне моделювання, мульти-агентні системи, моделювання керування, агентне моделювання, розподілені агентні системи

The paper covered the fundamental simulation aspects for automation and control problems in technical systems and complexes. Standards and existing protocols were reviewed to building such systems. Development of the simulation were investigator in scope of information systems synthesis, structural design, considering the connection with artificial intelligence systems. Modern software frameworks and modeling problems classification were analysed in the paper in scope of building complex control systems. The major part in the paper show the advantages of using distributed approach for modeling using the self organised agents and tasks decomposition. Determined the basic characteristics of simulation systems and justified prospects of using the distributed, scalable systems based on multi-agent approach. As a conclusion we demonstrate the intuitive clue for solving different kind of many practical problems from wide range of science.

Keywords: simulation, multi-agent systems, management modeling, agent-based modeling, distributed agent systems.

Введение. Большинство реальных систем в окружающем нас мире настолько сложны, что построение чисто аналитических моделей для таких систем – сверхсложная, не окупающая усилий задача. Для моделирования таких систем уместно применить имитационное моделирование (ИМ), особенностью которого является описание модели, в большей степени, в виде алгоритмов взаимодействия составных её частей [1]. Отчасти такой подход чем-то схож с человеческим видением окружающего мира.

В данный момент ИМ, как научное направление бурно развивается благодаря необходимости, а также возможности его применения для решения сложных и больших задач. Возможность использования ИМ для решения таких задач обусловлена двумя основными факторами. Во-первых, это – резкий рост вычислительных мощностей благодаря технологическому прорыву в микро и нано электронике, а во-вторых, стремительное развитие мировой грид-инфраструктуры, технологий линий связи (таких, как спектральное уплотнение оптических каналов WDM [2]) и виртуальных организаций. В-третьих, помимо фундаментальных исследований, при моделировании сложных процессов в физике, генной инженерии, биологии, астрономии, актуальность ИМ определяется огромными потребностями прикладного характера. Это вопросы, связанные с прогнозированием поведения сверхсложных систем, таких как, прогнозирование погодных условий в масштабах отдельно взятой страны или всего земного шара, социальных процессов (динамика численности населения), экономических процессов развития отраслей, фондового рынка и т.д. Поэтому развитие данного направления представляется крайне важным для решения насущных научных и прикладных задач.

История моделей и методов имитационного моделирования. ИМ, как направление в науке, зародилось в 1950-х годах с появлением первых ЭВМ, и было впервые применено в оборонной промышленности [3]. Первые имитационные модели разрабатывались с использованием обычных языков программирования общего назначения, таких, как FORTRAN, и были предназначены для использования на больших универсальных ЭВМ – мейнфреймах (англ. Mainframes) [4]. К началу 1960-х были созданы первые подходы и первые специальные языки имитационного моделирования (например, SIMSCRIPT и GPSS) [5], что способствовало распространению использования ИМ не только в оборонной промышленности, но и в других отраслях, например, в производственной и финансовой. Широкое распространение и доступность персональных компьютеров в начале 1980-х стали новым этапом в развитии технологии ИМ, которая стала использоваться еще более широким кругом специалистов, были созданы новые языки ИМ, такие как SLAM и SIMAN [6], предназначенные для использования на персональных компьютерах.

В конце 1980-х – ранних 1990-х новые графические возможности персональных компьютеров позволили разработчикам программного обеспечения (ПО) создать графические инструменты разработки моделей и использовать анимацию в ИМ. Например, такие инструменты ИМ, как ProModel [7], Arena [8], и iThink [9],

обеспечили управляемый в режиме меню пользовательский интерфейс, визуальное интерактивное моделирование и впечатляющие по тем временам возможности анимации. Эти усовершенствования еще больше упростили разработку имитационных моделей и значительно повлияли на широкое использование ИМ.

Примерно в то же время в разработке ПО стали активно применяться объектно-ориентированные подходы и анализ. Это, в свою очередь, не могло не оказать влияния на ИМ по причине существования природной связи между объектно-ориентированным моделированием и имитационным моделированием. Тогда же был введен новый термин – объектно-ориентированное имитационное моделирование (англ. object-oriented simulation). Объектно-ориентированное программирование, как технология, обеспечило новый скачок в ИМ, т.к. теперь программисты могли использовать разработанные ранее библиотеки объектов и существенно повысить скорость и качество разработки ПО в области ИМ. На этом этапе развития были разработаны специализированные объектно-ориентированные языки имитационного моделирования, например, Simple++, OOSimL, Simscript III [10]. Следует отметить, что у разработчиков, использующих эти языки, отпала необходимость в решении низкоуровневых задач при создании новых продуктов – вся необходимая логика уже содержалась в предоставляемых классах стандартных библиотек.

Параллельно, начиная с 1980-х годов, развитие сетевых технологий внесло свой вклад в развитие ИМ. Тогда же начались разработки распределенных систем ИМ, что подтолкнуло разработчиков к созданию новых стандартов, поскольку они были необходимы для корректного взаимодействия распределенных модулей. Разработки таких систем и стандартов не прекращаются до наших дней, а необходимость в них только растет. За последние 20-30 лет произошло еще одно качественное изменение в области ИМ. Был создан и устоялся новый подход в ИМ – мультиагентное (агентное) моделирование (АМ). В последнее время наблюдается тенденция в разработке систем ИМ, направленная на изменение их интерфейса с похожего на оболочку языка программирования к виду более понятному и удобному для пользователя. Это объясняется тем, что ИМ находит все более широкое применение в областях, далеких от компьютерных наук, и исследователи в этих областях, как правило, не владеют навыками программирования. Тем не менее, разработчики оставляют возможность написания частей модели на языке программирования, поскольку это позволяет обеспечить необходимую гибкость и охватить большее количество моделей.

Таким образом, ИМ, начав свое становление и развитие с середины XX века, вначале применялось для сравнительно небольшого спектра задач. Однако на данный момент методы и программные продукты ИМ используются практически во всех предметных областях, а исследования и разработки в этом направлении продолжают.

Задачи имитационного моделирования. ИМ широко применяется в самых различных отраслях человеческой деятельности. Примерами могут послужить: научные исследования в естественных и гума-

нитарных науках [11, 12], исследования в оборонной промышленности [13], проектирование непроизводственных и производственных систем [14, 15], систем массового обслуживания [16], логистических систем [17], создание тренажеров для специалистов различных отраслей [18], а также многие другие. Каждая из перечисленных категорий включает множество различных реальных систем. Например, теория систем массового обслуживания включает такие системы, в которых могут присутствовать очереди, конкуренция за ресурсы, отказы в обслуживании в случае, если необходимый ресурс занят или отсутствует. Примером таких систем могут послужить: магазины, распределенные вычислительные системы, компьютерные сети и так далее. Такое обилие моделей можно классифицировать с точки зрения математического моделирования.

В ИМ устоялось три основных подхода [22]: системная динамика, дискретно-событийное моделирование и агентное моделирование, которые перечислены в хронологическом порядке их создания. Каждый из подходов имеет свою терминологию, свои методологии и стандарты. Современные программные продукты ИМ разрабатываются на основе одного или одновременно нескольких из этих подходов.

Системную динамику [23, 24] основал Джей Форрестер в 1950-х годах. Этот подход был создан раньше остальных в ИМ и из названия можно понять, что используется он, в большей степени, для исследования динамических систем. Взгляд на исследуемые системы с точки зрения системной динамики называют «системным мышлением» [25].

В системной динамике существует своя устоявшаяся терминология. Так, например, модели в системной динамике задаются с помощью накопителей (агрегатов) и потоков между ними, а также правил (обратных связей) и задержек [22, 26]. Накопитель (агрегат) – переменная, которая изменяется во времени и хранит в себе информацию о количественной характеристике моделируемой системы. Поток характеризуется его величиной и влияет на скорость этих изменений, а обратные связи могут быть как положительными, так и отрицательными, и вместе с задержками влияют на величину потока. С помощью системной динамики строят экономические модели (модели бизнес-процессов, развития города, производства), динамики популяции, экологии, развития эпидемии и множество других подобных моделей.

Дискретно-событийное моделирование предложено профессором Джеффри Гордоном в 1960х годах. Его особенность также подчеркнута в его названии и состоит в том, что функционирование системы представляется как хронологическая последовательность событий [27]. Поэтому один из центральных технических вопросов в программировании дискретно-событийных моделей – написание планировщика, который извлекает из списка запланированных событий событие с минимальной временной меткой и выполняет его, а также добавляет в этот список новые запланированные события. Этот подход очень развит и применяется для исследования широкого спектра имитационных моделей.

Агентное моделирование последнее время становится все более популярным [28]. В этом подходе исходными данными для модели является информация о взаимодействии ее составных частей (агентов) между собой и окружающим миром, а на выходе исследователь такой модели получает информацию о глобальных закономерностях, ей присущих. Под агентом понимается сущность, которая способна воспринимать воздействия внешнего мира и сама оказывать воздействия на него [28]. Кроме того, среди прочих характеристик агента подразумевается его способность самостоятельно функционировать, адаптироваться к изменениям окружающего мира, выполнять поставленные перед ним цели и т.д. Этот подход в ИМ тесно переплетается с искусственным интеллектом, и поэтому очень часто в литературе употребляется термин «интеллектуальный агент» вместо термина «агент». «Интеллект» агента может варьироваться от простого рефлекторного ответа на влияния внешнего мира до способности обучаться. Как правило, реальные системы невозможно исследовать с помощью одного агента – такой подход не обеспечит необходимый уровень конкретизации, поэтому необходимо использовать значительное количество агентов, взаимодействующих друг с другом. По этой причине системы моделирования, использующие агентный подход, часто называют мультиагентными системами (МАС) [29].

На сегодняшний день, АМ – мощная технология, предназначенная для исследования сложных систем. С термином «агентное моделирование» тесно связан термин «агентно-ориентированное программирование». Успех АМ в значительной степени принесло свойство агентных моделей охватывают системы с различными уровнями абстракции (уровень абстракции выбирается от самого высокого до низкого на усмотрение исследователя системы). Также в одной агентной модели можно различным компонентам определять различные уровни абстракции. Например, один агент может представлять отдельного человека, а другой – целое отделение предприятия.

Таким образом, рассмотрев основные подходы в имитационном моделировании, можно сделать вывод, что каждый из них предназначен для решения своего круга задач. Однако для исследований с помощью ИМ необходимо, но недостаточно знать основные подходы. Для своей работы исследователь дополнительно должен знать и использовать специализированные стандарты, определенные методологии, концептуальные фреймверки, специализированные языки программирования и так далее. Уже существует значительное количество подобных решений, которые тяжело охватить в рамках одной публикации. Поэтому дальнейшее рассмотрение будем проводить с уклоном на агентно-ориентированные решения, которые основаны, как нам кажется, на наиболее перспективных идеях с точки зрения реализации их на распределенных вычислительных сетях типа GRID.

Медународные стандарты систем ИМ. С развитием ИМ возникла необходимость решать сложные задачи, связанные с разработкой специальных программных средств. В частности это: повторное использование кода существующих моделей, вопросы их взаимодействия (причем они могут быть распо-

жены как на одном компьютере, так и на различных компьютерах в сети) и т.д.

В начале 80-х годов с развитием компьютерной техники возникла возможность создавать распределенные системы имитационного моделирования, следовательно и необходимость стандартизировать разработку таких систем, которые уже не ограничиваются отдельной вычислительной машиной, а функционируют в рамках вычислительных сетей. Поэтому стал вопрос осуществления взаимодействия уже существующих моделей по некоторым правилам. Это был новый качественный этап в развитии систем имитационного моделирования. Самые известные стандарты в распределенном имитационном моделировании: High Level Architecture, Distributed Interactive Simulation, Common Image Generator Interface [30] и другие.

High Level Architecture. High Level Architecture (HLA) (IEEE 1516) [31] – это стандарт архитектуры распределенных систем имитационного моделирования, который первоначально разрабатывался для военных потребностей. Его разработчик – Department of Defense, Defense Modeling and Simulation Office (DoD DMSO). HLA стандарт был разработан, чтобы способствовать функциональной совместимости и осуществлению взаимодействия различных составляющих военных систем: имитационных моделей, датчиков тренажеров, датчиков на реальной боевой технике. Благодаря HLA, имитационные модели могут взаимодействовать друг с другом, будучи используемыми на различных компьютерах, не зависимо от компьютерных платформ. Коммуникация между моделями осуществляется с помощью специально разработанной библиотеки RTI (Run-Time Infrastructure). Существуют версии RTI, написанные на языке программирования C++ или Java, но она также может быть использована при написании программ и на других языках программирования.

В стандарте HLA существует своя терминология: федератом называется отдельная составляющая всей системы (например, имитационная программа, отдельный датчик, тренажер и т.д.). Объединение федератов – это федерация. Можно провести аналогию данного стандарта с агентным подходом: федерат можно сопоставить понятию агента, а федерацию – мультиагентной системе.

Distributed Interactive Simulation. Стандарт Distributed Interactive Simulation (DIS) (IEEE 1278) [32] так же, как и HLA, первоначально проектировался для военных целей. Но исторически стандарт DIS был разработан раньше, чем HLA. DIS – это открытый стандарт для сопровождения взаимодействия моделей, существующих в режиме реального времени, на распределенных системах. Из-за того, что разные имитационные программы написаны на разных языках или диалектах одного языка, протокол DIS был разработан, чтобы способствовать взаимодействию различных участников сети. В отличие от HLA, этот стандарт определяет только стандарт информационных сообщений имитационных приложений и не затрагивает внутреннего представления моделей.

Протокол DIS гарантирует безопасность и точность передачи сообщений между множеством имитационных программ, объединенных в одну сеть. DIS кодирует информацию имитационных программ в форматированные сообщения, которые называются Protocol Data Units (PDUs), которыми обмениваются хосты с использованием существующих протоколов транспортного уровня, таких как User Datagram Protocol (UDP). Имитационное приложение должно поддерживать определенный тип PDU, чтобы обеспечить отправку/прием этой информации другому приложению.

тационных программ, объединенных в одну сеть. DIS кодирует информацию имитационных программ в форматированные сообщения, которые называются Protocol Data Units (PDUs), которыми обмениваются хосты с использованием существующих протоколов транспортного уровня, таких как User Datagram Protocol (UDP). Имитационное приложение должно поддерживать определенный тип PDU, чтобы обеспечить отправку/прием этой информации другому приложению.

Common Image Generator Interface. Стандарт Common Image Generator Interface (CIGI) [33] пока не является общепризнанным мировым стандартом, но, тем не менее, на него стоит обратить внимание. Этот стандарт зародился в компании Boeing, которая уже несколько десятков лет успешно применяет имитационное моделирование для создания тренажеров летательных аппаратов.

Это открытый стандарт и предназначен он для трансляции информации от сервера с программой имитации к устройству отображения информации и последующей визуализации. Архитектурно он представляет собой протокол передачи пакетов и API, который эти пакеты формирует со стороны сервера, принимает и распаковывает со стороны устройства визуализации. В своей основе он также как и DIS использует протокол UDP.

The Foundation for Intelligent Physical Agents. Дальнейшее развитие метода имитационного моделирования потребовало создание новых стандартов. В 2005 году организацией The Foundation for Intelligent Physical Agents (FIPA) [34–36] был принят ряд стандартов, касающихся самого нового и перспективного подхода – агентного моделирования. Они предназначены для продвижения агентно-ориентированной технологии и взаимодействия ее с другими технологиями. Эти стандарты регламентируют разработку как отдельных агентов, так и мультиагентных систем в целом. В частности, эти стандарты, так же, как HLA и DIS, регламентируют правила создания распределенных мультиагентных систем.

Методологии проектирования систем ИМ. Так как задача проектирования сложных систем далеко не тривиальна, возникла необходимость в создании методологий для этих целей. Здесь под термином «методология», как и в объектно-ориентированном программировании, имеется в виду последовательность действий при разработке архитектуры программы. Самые известные методологии проектирования агентно-ориентированных систем: Gaia, Agent UML, MaSE, MESSAGE, Tropos, Prometheus [37].

Gaia. Методология Gaia [38] была разработана Михаэлем Вулдриджем (Michael Wooldridge), Франко Замбонелли (Franco Zambonelli) и Николасом Дженнингсом (Nicholas Jennings). Она представляет собой инструмент для проектирования архитектуры мультиагентных систем. Gaia заимствует ряд принципов и понятий из объектно-ориентированного анализа и синтеза. Тем не менее, она не является попыткой применить объектно-ориентированный подход для создания мультиагентных систем, а предоставляет разработчику принципиально новый инструмент. И поэтому она позволяет решать такие задачи разработ-

ки агентно-ориентированных систем, как охват гибкости агентов, автономное поведение и решение задач, взаимодействие агентов и так далее, что не предусматривает объектно-ориентированный анализ и синтез.

Gaia рассматривает MAC как концептуальную организацию, состоящую из взаимодействующих агентов с различными ролями. Один агент может агрегировать в себе несколько ролей. В соответствии идеологии Gaia, агенты предоставляют сервисы, которые соответствуют их ролям. Процесс разработки в Gaia состоит из двух этапов: анализа (analysis) и проектирования (design). Проходя эти этапы, разработчик последовательно переходит от абстрактных сущностей к более конкретным. К недостаткам Gaia можно отнести то, что она накладывает ряд существенных ограничений на разрабатываемые модели, что значительно сужает класс мультиагентных систем, которые могут быть разработаны с ее применением.

Agent UML. Все более широкое применение в исследовании и разработке мультиагентных систем приобретает язык графической объектной разработки приложений – UML [39] (Unified Modeling Language – Унифицированный Язык Моделирования). Язык UML является языком общего назначения, поэтому его можно применять для решения достаточно широкого спектра задач. Например, названный выше стандарт FIPA использует UML.

Методология, использующая UML для разработки архитектуры мультиагентных систем, называется Agent UML (AUML) [40]. Поскольку UML используется для объектно-ориентированной разработки, то данная методология рассматривает агентно-ориентированный подход как расширение объектно-ориентированного. Разработчики Agent UML максимально используют функциональность UML и разрабатывают только те нововведения, которые никак нельзя заменить существующими в UML возможностями. До недавнего времени разработчики вкладывали значительные усилия для активного развития этой методологии, но сейчас ее развитие приостановлено, поскольку в нем нет необходимости – требуемая функциональность уже частично или полностью реализована в последних версиях UML. Например, версия UML 2.1 была выпущена с множеством функциональных возможностей, которые предполагалось осуществить разработчиками AUML. Также существуют попытки расширить другие методологии, в том числе и методологию Gaia с использованием UML [41]. Выше были рассмотрены две наиболее перспективные, по нашему мнению, методологии построения систем ИМ. С остальными методологиями можно познакомиться в источниках [42–44].

Программные продукты. На основе рассмотренных выше подходов, стандартов и методологий создаются программные продукты для проведения исследований с помощью имитационного моделирования.

На сегодня существует множество различных программных продуктов для работы с имитационными моделями – систем имитационного моделирования, готовых фреймворков для создания моделей и самих систем моделирования, специальных библиотек

для различных языков программирования, библиотек и надстроек для математических пакетов и т.д. Они подразделяются на специализированные и общего назначения, ориентированные на последовательные или параллельные вычисления, поддерживающие различные подходы и различные алгоритмы продвижения времени.

Проанализируем наиболее известные из существующих современных программных продуктов. Анализ начнем с агентно-ориентированных решений, а закончим наиболее популярными продуктами, использующими другие подходы – GPSS World, AnyLogic, Simulink.

NetLogo [45] – свободно распространяемое кроссплатформенное окружение для программирования мультиагентных систем. Позволяет создавать распределенные решения, путем взаимодействия моделей в сети через использование технологии [HubNet](#) [46]. Ее разработал Uri Wilensky в 1999 году. На сегодняшний день работу над этим фреймворком продолжают в центре The Center for Connected Learning and Computer-Based Modeling. NetLogo применяется в основном для исследования в естественных и социальных науках [47]. К преимуществам NetLogo можно отнести: кроссплатформенность, высокую функциональность моделей, наличие мощных инструментов визуализации, возможность использования в Web, а также взаимодействия моделей по сети, наличие специального инструмента для сбора статистических данных о тестировании модели с различными значениями параметров.

Кроссплатформенность обеспечивается за счет того, что среда NetLogo написана на языке Java. Поэтому ее можно использовать в различных операционных системах: Mac, Windows, Linux и т.д.

NetLogo использует специальный язык программирования, что в свою очередь обеспечивает широкую функциональность моделей. Кроме того, этот язык специализирован для создания агентных моделей и содержит большую библиотеку примитивов, что ускоряет разработку моделей. Но, кроме того, NetLogo имеет встроенный редактор интерфейса моделей.

VisualBots [48] – бесплатная библиотека для Visual Basic, предназначенная для создания визуальных мультиагентных систем. Она написана специально для использования в Microsoft Excel, что позволяет использовать функциональность этого приложения. Содержит обширный набор примеров моделей. Это в свою очередь позволяет достаточно быстро научиться создавать свои собственные модели.

Несомненным преимуществом рассматриваемой библиотеки является продуманный и удобный графический пользовательский интерфейс для запуска моделей, созданных на ее основе. Она позволяет создавать обширные объектные модели с большими графическими возможностями, а также управлять группами агентов и содержит алгоритмы быстрого поиска соседних агентов. Особенность VisualBots в том, что данная разработка предназначена для построения и исследования визуальных моделей. Это значительно сужает количество возможных моделей, что можно отнести к недостаткам данной разработки.

MASON [49] представляє собою Java-бібліотеку для розробки та моделювання мультиагентних систем і повністю підтримує мову Java, починаючи з версії 1.3 і вище. Програми написані з використанням даної бібліотеки мобільні за рахунок використання мови Java, а також порівняно невеликого розміру. Основними перевагами MASON можна назвати кроссплатформеність і наявність потужних інструментів візуалізації, яка може бути додана, видалена або змінена в будь-який момент часу. Підтримується двохмерна і трьохмерна візуалізація. Крім того, існує можливість зберігати зображення, відео процесу імітації, графіки та потоки вихідних даних.

REPASt. Бібліотека REPASt (Recursive Porus Agent Simulation Toolkit) [50] представляє собою набір інструментів для створення моделей мультиагентних систем. Цей фреймворк – безкоштовний і відкритий проєкт. Більш ранні версії повністю підтримували мови програмування Java та C#. Також моделі Repast можна розробляти на різних мовах, включаючи: C++, Visual Basic.Net, Lisp, Prolog, та Python.

Остання версія включає в себе створення гібридних моделей з використанням мови програмування Java [51], Groovy [52] та блок-схем моделей, має зручний користувацький інтерфейс, а також потужну систему візуалізації, яка дозволяє легко створювати та візуалізувати 2D та 3D моделі.

В даній середі інтегровано велике число стандартних бібліотек, наприклад: бібліотека Weka [53], призначена для отримання нових знань, використовуючи алгоритми машинного навчання; бібліотеки для генетичних алгоритмів, нейронних мереж; методи регресії; бібліотеки генерації випадкових чисел, та бібліотеки спеціалізованої математики. Крім переваг REPASt можна віднести: підтримується запис та вилучення результатів експериментів з XML-файлів, текстових файлів та бази даних; повний багатопотоковий дискретно-подієвий планировщик; автоматизований фреймворк моделювання з використанням методу Монте-Карло, який підтримує роботу з кількома моделями одночасно та оптимізацію результатів; повністю об'єктно-орієнтована середа. Також REPASt містить вбудовані інструменти для інтеграції зовнішніх моделей. Доступні версії REPASt для всіх популярних платформ, включаючи Windows, Mac OS, Linux.

Java Agent Development Framework (JADE) [54] – Java-бібліотека для створення розподілених мультиагентних систем. Найважливішою перевагою JADE – це те, що створювані моделі відповідають розглянутому вище стандарту FIPA (The Foundation for Intelligent Physical Agents). Оскільки JADE розроблена з використанням мови Java, то вона успадковує від неї всі переваги та недоліки.

Розглянуті вище програмні засоби в основному орієнтовані на використання АМ та відповідні задачі. Однак достатньо широкий

клас задач ними не вирішується, тому слід звернути увагу на інше програмне забезпечення, що використовує інші підходи.

GPSS World. В основі GPSS World покладено мову GPSS (General Purpose System Simulation – загальноцільова система моделювання), яку розробив співробітник фірми IBM професором Джефрі Гордоном в 1961 році, засновником дискретно-подієвого підходу в імітаційному моделюванні [55]. GPSS World – система загального призначення, яку можна використовувати для розв'язання широкого спектра задач, підтримує як неперервні так і дискретні моделі. В ній існує можливість моделювати непродуктивні, продуктивні системи, системи масового обслуговування. Особливістю GPSS World є те, що для її використання необхідно вивчити спеціальну мову програмування.

AnyLogic. Розробчик – компанія JI Technologies. AnyLogic є середою імітаційного моделювання загального призначення та підтримує розробку великої кількості типів моделей, а також всі три основні підходи, що існують в імітаційному моделюванні. Дозволяє будувати гібридні моделі з одночасним використанням кількох підходів. Система має графічний інтерфейс користувача, але для забезпечення додаткової гнучкості розробляються моделі, AnyLogic дозволяє додатково використовувати мову програмування Java для створення моделей.

SIMULINK. Деякі розробники програмного забезпечення пішли шляхом створення інструментів (бібліотек) імітаційного моделювання, які включаються в прикладні математичні пакети. Таким прикладом може послужити середою візуального моделювання SIMULINK, яке включено в стандартний набір математичного пакета MATLAB [56].

Розподілене мультиагентне моделювання. Багато моделей імітаційного моделювання своєю природою досить добре підходять для паралельних обчислень. В більшій мірі це можна сказати про мультиагентні, а також дискретно-подієві моделі. Даний питання стає важливим, оскільки з допомогою мультиагентних моделей можна досліджувати складні глобальні системи, які можуть включати в себе від кількох тисяч до кількох мільйонів агентів, що вимагає значительних обчислювальних ресурсів. Тому великомасштабний імітаційний експеримент на локальній однопроцесорній обчислювальній машині займе неприпустимо довгий час. Або ж проведення такого експерименту взагалі буде неможливим через недостатність пам'яті локальної обчислювальної машини. Отже, виникає необхідність використання обчислювальних архітектур, які дозволяють привертати значительне число різних ресурсів (обчислювальні ресурси, пам'ять та т.д.). З іншої сторони, перед розробником системи мультиагентного моделювання, орієнтованої на паралельні та розподілені обчислення, виникають додаткові задачі, які він повинен вирішити. Сюди можна віднести, наприклад, синхронізацію модельного часу [57], дина-

мическое распределение ресурсов и так далее. Grid-технология позволяет освободить разработчика если не от всех, то от большинства из этих задач посредством использования middleware [58]. Уже одно это обстоятельство делает распределенное мультиагентное моделирование на Grid более привлекательным, чем использование других вычислительных архитектур. Кроме того, Grid-технологии позволяют использовать значительное количество ресурсов для проведения эксперимента. По этим причинам все большей актуальности и внимания среди исследователей набирает вопрос мультиагентного моделирования именно на Grid-системах. Из рассмотренных выше существующих решений распределенные мультиагентные системы позволяют создавать: NetLogo, Mason, JADE, но они не ориентированы на Grid. К современным распределенным мультиагентным системам имитации, как и к распределенным системам имитации в целом, выдвигаются требования гибкости, прозрачности, масштабируемости [58].

Все перечисленные требования можно осуществить только при условии правильного выбора архитектуры системы моделирования. Как показывает практика, в большинстве случаев наиболее разумная общая архитектура систем моделирования на Grid следующая:

1. В основе системы лежит Grid middleware, предоставляющий свои сервисы.
2. Над первым уровнем находится имитационный «движок», который предоставляет все необходимые сервисы следующему уровню (например, мониторинг состояния агентов, синхронизация времени и т. д.)
3. Самый верхний уровень – прикладной, который представляет собой конкретные имитационные программы (эксперименты), использующие сервисы движка.

Существует два пути реализации первого уровня:

1. Использование уже существующего Grid middleware.
2. Создание собственного Grid middleware специально для этих целей.

В данном случае архитектура приложения является сервисно-ориентированной (англ. *SOA, service-oriented architecture*), поскольку каждый уровень использует сервисы нижележащего уровня, а пользователь использует сервисы самого верхнего уровня [59]. Использование этой архитектуры дает следующие основные преимущества:

- улучшение понимания структуры приложения;
- упрощение взаимодействия его составляющих частей;
- возможность создания кроссплатформенных приложений;
- повышение масштабируемости и повторного использования кода разрабатываемой системы.

Следует отметить, что использование именно сервисно-ориентированной архитектуры естественным образом подходит для мультиагентного моделирования и отображено в существующих стандартах и методологиях. Например, рассмотренная выше методология разработки мультиагентных систем Gaia предполагает, что каждый агент может как использо-

вать сервисы других агентов, так и самостоятельно предоставлять эти сервисы.

Попробуем классифицировать существующие концепты инструментариев для мультиагентного моделирования на Grid.

Первая группа. Было предпринято значительное количество попыток использовать стандарт HLA для создания систем мультиагентного моделирования на Grid путем взаимодействия различных моделей согласно этому стандарту. В таком случае особенностью разрабатываемых систем можно считать строгую иерархическую древовидную модель, а HLA можно рассматривать как централизованный механизм координации распределенных ресурсов моделирования. Самым известным из подобного решения можно назвать HLA_Grid_RePast. Система разработана на основе двух независимых платформ: HLA_Grid и HLA_RePast. Первая предназначена для поддержки стандарта HLA на Grid, вторая – для распределенного мультиагентного моделирования путем взаимодействия моделей в соответствии со стандартом HLA. Так же известны: XMSF, HLA+Open GIS Consortium (OGC), HLA-based Interactive Simulation on the Grid, IDSim. Использует Globus Toolkit, SO-XMSSEA.

Вторая группа. Есть попытки расширения обычных (нераспределенных) систем мультиагентного моделирования для использования на Grid. Сюда можно отнести упомянутый выше HLA_Grid_RePast, а также Distributed GenSim и MACE-3J. Например, MACE-3J разворачивается с использованием Globus Toolkit 3 (GT3).

Третья группа. Также существуют попытки адаптации под Grid систем для распределенного имитационного моделирования, не предназначенных для Grid изначально. Автор Iskra предложил Time Warp Simulation Kernel – средство, которое позволяет работать в Grid-среде существующим системам моделирования с использованием оптимистического алгоритма продвижения времени.

Четвертая группа. Системы, ориентированные на движок продвижения времени. Они реализованы с использованием классических алгоритмов продвижения времени в распределенном имитационном моделировании (консервативный и оптимистический алгоритм). Архитектура систем этой группы может значительно отличаться от описанной выше. Например, в JAMES использован явно централизованный событийно-ориентированный подход, а MPADES использует консервативный алгоритм продвижения времени с централизованной моделью среды.

Пятая группа (остальные). **Large Scale Multiagent Simulation on the Grid** (автор Ingo J. Timm). Использует технологию P2P-Grid, реализованную посредством проекта JXTA. JXTA содержит стек XML-ориентированных протоколов для взаимодействия по принципу P2P (peer-to-peer, «равный к равному»). Ядро JXTA реализует сервисы Grid, такие как поиск, индексация, группировка и безопасность. JXTA не зависит от нижележащих протоколов транспортного уровня и реализует необходимую для Grid масштабируемость и децентрализацию. На прикладном уровне в этой системе используется JADE.

Выводы. В настоящее время в развитии систем имитационного моделирования продолжают тенденции использования распределенной архитектуры, разработки графических инструментов для создания имитационных моделей, внедрения мощных инструментов визуализации имитационных экспериментов. Создаются новые специализированные языки программирования (например, агентно-ориентированный язык программирования Axum [60], который разрабатывается в лаборатории DevLabs корпорации Microsoft). Также продолжают теоретические исследования в этой области, развитие существующих и создание новых методологий и стандартов. Это закономерно, поскольку имитационное моделирование уже давно подтвердило свою эффективность в качестве мощного инструмента для исследования сложных систем. Тем не менее, еще остается множество нерешенных задач и проблем.

Как показывает практика, современная среда распределенного мультиагентного моделирования для научных исследований должна соответствовать следующим требованиям: наглядность и простота задания агентов; масштабируемость – ограничения, накладываемые на модели (например, максимально допустимое количество агентов в модели) должны в большей степени определяться только возможностями аппаратного обеспечения; при разработке модели пользователь не должен решать концептуальные проблемы такие, как синхронизация времени, обмен сообщениями между агентами, функционирование окружающего мира и т.д.; соответствие современным понятиям об архитектуре системы (модульность, сервисно-ориентированность и т.д.); возможность дальнейшего усовершенствования; ориентация на Grid, как на наиболее мощный и доступный инструмент распределенных вычислений.

Проведенный анализ показал, что среди существующих решений нет таких, которые в полной мере удовлетворяли бы данную потребность. Это позволяет сделать вывод, что в настоящее время существует необходимость в создании масштабируемого высокоуровневого инструмента мультиагентного моделирования, обладающего мощным графическим интерфейсом для создания моделей и визуализации процесса и результатов моделирования. Данный инструмент мог бы успешно применяться при решении большого количества научных и прикладных задач высокой степени сложности, состоящих из большого количества компонентов, связи между которыми не являются очевидными, а порой и непредсказуемыми.

Список литературы:

1. *Лычкина, Н. Н.* Имитационное моделирование экономических процессов : Учеб. пособие [Текст] / Под. ред. В. В. Година. – М.: Академия ИТ, 2005. – 165 с.
2. *Keiser, G. E.* A Review of WDM Technology and Applications [Text] / G. E. Keiser // Opt. Fiber Technol. – 1999. – Vol 5 (1), 3.
3. *Harrington, H. J.* Simulation modeling methods. To Reduce Risks and Increase Performance [Text] / H. J. Harrington // McGraw-Hill Professional, 2000. – 379 p.
4. Mainframe computer – Wikipedia, the free encyclopedia [Электронный ресурс]. Режим доступа: http://en.wikipedia.org/wiki/Mainframe_computer. – Загл. с экрана.
5. *Mikihiko, O.* Simulation engineering [Text] / O. Mikihiko // IOS Press Amsterdam, 1998. – 190 p.
6. *Fishman, G. S.* Discrete-event simulation: modeling, programming, and analysis [Text] / G. S. Fishman // Springer, 2001. – 537 p.
7. Decision Making Tools and Techniques – ProModel [Электронный ресурс]. Режим доступа: <http://www.promodel.com/>. – Загл. с экрана.
8. Arena Portal: Products [Электронный ресурс]. Режим доступа: http://www.arenasimulation.com/Products_Products.aspx. – Загл. с экрана.
9. Isee systems – Ithink Modeling & Simulation Software for Business [Электронный ресурс]. Режим доступа: <http://www.iseesystems.com/software/Business/ithinkSoftware.aspx>. – Загл. с экрана.
10. *Garrido, J. M.* Object Oriented Simulation: A Modeling and Programming Perspective [Text] / J. M. Garrido // Springer, 2009. – 449 p.
11. *Heermann, D. W.* Computer Simulation Methods in Theoretical Physics [Text] / D. W. Heermann // Springer-Verlag Berlin Heidelberg, 1990. – 155 p.
12. *Hartmann, S.* The World as a Process: Simulations in the Natural and Social Sciences [Text] / S. Hartmann // Modelling and Simulation in the Social Sciences from the Philosophy of Science Point of View, Theory and Decision Library. – Dordrecht: Kluwer, 1996. – P. 77–100.
13. Introduction to Military Training Simulation: A Guide for Discrete Event Simulationists [Электронный ресурс]. Режим доступа: <http://www.modelbenders.com/papers/mil4des.html>. – Загл. с экрана.
14. *Miller, S.* Introduction to manufacturing simulation/ Scott Miller, Dennis Pegden [Text] / S. Miller // Proceedings of the 2000 Winter Simulation Conference, 2000.
15. *Banks, J.* Handbook of simulation: principles, methodology, advances, applications, and practice, vol. 57 [Text] / J. Banks // Wiley-IEEE, 1998. – 849 p.
16. *Томашевский, В.* Имитационное моделирование в среде GPSS [Текст] / Томашевский В., Жданова Е. – М.: Бестселлер, 2003. – 416 с.
17. *Graudina, V.* Technologies and multi-agent system architectures for transportation and logistics support Vita Graudina, [Text] / V. Graudina, J. Grundspenkis // International Conference on Computer Systems and Technologies - CompSysTech, Varna, Bulgaria, 2005.
18. *Воронцов, К. В.* Имитационное моделирование торгов: новая технология биржевых тренажеров [Электронный ресурс] / К. В. Воронцов, С. Б. Пиеничников // Индикатор – 2002. – No2 (42) – Режим доступа: http://www.forecsys.com/ru/site/about/media_coverage/exchange_simulator/. – Заголовок с экрана
19. *Шеннон, П.* Имитационное моделирование систем: искусство и наука [Текст] / П. Шеннон. – М.: Мир, 1978.
20. *Карпов, Ю.* Имитационное моделирование систем. Введение в моделирование с AnyLogic 5 [Текст] / Ю. Карпов. – СПб.: БХВ-Петербург, 2005. – 400с.
21. *Ljung, L.* Modeling of dynamic systems [Text] / L. Ljung, T. Glad // Prentice Hall PTR, 1994. – 368 p.
22. *Борищев, А. А.* От системной динамики и традиционного ИМ – к практическим агентным моделям: причины, технология, инструменты [Электронный ресурс]. – Режим доступа: <http://www.gpss.ru/>;
23. *Forrester, J. W.* Principles of systems [Text] / J. W. Forrester // Wright-Allen Press Inc., 1968. – 387 p.
24. *Forrester, J.* Industrial Dynamics – A Major Breakthrough for Decision Makers [Text] / J. Forrester // Harvard Business Review. – 1958. – Vol. 36, No. 4. – P. 37–66.
25. *Richmond, B.* System Dynamics/Systems Thinking: Let's Just Get On With It: International Systems Dynamics Conference in Sterling, Scotland. [Text] / B. Richmond // Springer, 1994. – 215 p.
26. *Ruth, M.* Dynamic modeling for business management: an introduction [Text] / M. Ruth, B. Hannon // Springer, 2003 – 388 p.
27. *Banks, J.* Discrete-Event System Simulation [Text] / Jerry Banks, John S. Carson, Barry L. Nelson, David M. Nicol // Prentice Hall, 2009. – 622 p.
28. *Стюарт, Р.* Искусственный интеллект: современный подход [Текст] / Рассел Стюарт, Нораг Пумер, 2-е изд.: Пер. с англ. – М.: Издательский дом «Вильямс», 2006. – 1408 с.
29. *Shoham, Y.* Multiagent systems: algorithmic, game-theoretic, and logical foundations [Text] / Yoav Shoham, Kevin Leyton-Brown. – Cambridge University Press, 2008. – 483 p.
30. Industry Standards [Электронный ресурс]. Режим доступа: <http://www.mak.com/community/standards.php>. – Загл. с экрана.

31. Kuhl, F. Creating Computer Simulation Systems: An Introduction to the High Level Architecture [Text] / Frederick Kuhl, Richard Weatherly, Judith Dahmann. – Prentice Hall PTR, 1999. – 224 p.
32. DIANE Publishing Company. Distributed Interactive Simulation of Combat [Text]: DIANE Publishing Company. – DIANE Publishing, 1996. – 177 p.
33. CIGI: Overview [Электронный ресурс]. Режим доступа: <http://cigi.sourceforge.net/overview.php>. – Загл. с экрана.
34. The Foundation for Intelligent Physical Agents [Электронный ресурс]. Режим доступа: <http://fipa.org/>. – Загл. с экрана.
35. Subrahmanian, V. S. Heterogeneous agent systems [Text] / V. S. Subrahmanian // MIT Press, 2000. – 580 p.
36. Standard Status Specifications [Электронный ресурс]. Режим доступа: <http://fipa.org/repository/standardspecs.html>. – Загл. с экрана.
37. Давыдов, А. А. О Компьютационной теории социальных агентов [Электронный ресурс]. – Режим доступа: http://www.isras.ru/files/File/Socis/1-6-2006/davydov_agents.pdf.
38. Zambonelli F. Developing MAS: The Gaia Methodology [Text] / F. Zambonelli, N. Jennings, M. Wooldridge / ACM. – 2003. – Vol. 12, No 3.
39. Object Management Group – UML [Электронный ресурс]. Режим доступа: <http://uml.org/>. – Загл. с экрана.
40. Agent UML [Электронный ресурс]. Режим доступа: <http://www.auml.org/>. – Загл. с экрана.
41. Juan, C. Garcia-Ojeda. Extending the Gaia Methodology with Agent-UML [Text] / Juan C. Garcia-Ojeda, Jose de J. Perez-Alcazar, Alvaro E. Arenas / ACM. – 2004. – Vol. 3. – P. 1456–1457.
42. Schreiber, Th. CommonKADS: A comprehensive methodology for KBS development [Text] / A. Th. Schreiber, B. J. Wielinga, J. M. Akkermans, and W. Van de Velde // IEEE Expert. – 1994. – No 9(6).
43. The Prometheus Methodology [Электронный ресурс]. Режим доступа: <http://www.cs.rmit.edu.au/agents/SAC2/methodology.html>. – Загл. с экрана.
44. MESSAGE - Methodology for Engineering Systems of Software Agents [Электронный ресурс]. Режим доступа: <http://www.eurescom.de/public/projects/P900-series/p907/>. – Загл. с экрана.
45. NetLogo [Электронный ресурс]. Режим доступа: <http://ccl.northwestern.edu/netlogo/>. – Загл. с экрана.
46. Understanding HubNet [Электронный ресурс]. Режим доступа: <http://ccl.northwestern.edu/netlogo/hubnet.html>. – Загл. с экрана.
47. What is NetLogo? [Электронный ресурс]. Режим доступа: <http://ccl.northwestern.edu/netlogo/docs/>. – Загл. с экрана.
48. VisualBots Software [Электронный ресурс]. Режим доступа: <http://www.visualbots.com/software.htm>. – Загл. с экрана.
49. MASON Multiagent Simulation Toolkit [Электронный ресурс]. Режим доступа: <http://www.cs.gmu.edu/~eclab/projects/mason/>. – Загл. с экрана.
50. Repast Agent Simulation Toolkit [Электронный ресурс]. Режим доступа: <http://repast.sourceforge.net/>. – Загл. с экрана.
51. Java [Электронный ресурс]. Режим доступа: <http://java.com/>. – Загл. с экрана.
52. Groovy – Documentation [Электронный ресурс]. Режим доступа: <http://groovy.codehaus.org/Documentation>. – Загл. с экрана.
53. Weka 3 – Data Mining with Open Source Machine Learning Software in Java [Электронный ресурс]. Режим доступа: <http://www.cs.waikato.ac.nz/ml/weka/>. – Загл. с экрана.
54. Jade – Java Agent Development Framework [Электронный ресурс]. Режим доступа: <http://jade.tilab.com/>. – Загл. с экрана.
55. Кудрявцев, Е. М. GPSS World. Основы имитационного моделирования различных систем [Текст] / Е. М. Кудрявцев. – М.: ДМК Пресс, 2004. – 320 с.
56. Гуляев, А. МАТЛАВ 5.2. Имитационное моделирование в среде Windows [Текст] / А. Гуляев. – М.: Корона принт, 1999. – 288 с.
57. Окольников, В. В. Представление времени в имитационном моделировании [Текст] / В. В. Окольников // Вычислительные технологии. – 2005. – Том 10, No5.
58. Timm, I. J. Large Scale Multiagent Simulation on the Grid [Text] / Ingo J. Timm, Dirk Pawlaszczyk // IEEE International Symposium on Cluster Computing and the Grid (CCGRID), Cardiff University, Cardiff, UK, 2005.
59. OASIS SOA Reference Model TC [Электронный ресурс]. Режим доступа: http://www.oasis-open.org/committees/tc_home.php?w_g_abbrev=soa-rm. – Загл. с экрана.
60. Ахун [Электронный ресурс]. Режим доступа: <http://msdn.microsoft.com/en-us/devlabs/dd795202.aspx>. – Загл. с экрана.

Bibliography (transliterated):

- Lychkina, N. N. (2005). Imitacionnoe modelirovanie ekonomicheskix processov : Ucheb.posobie. Moscow: Akademiya IT, 165.
- Keiser, G. E. (1999). A Review of WDM Technology and Applications. Opt. Fiber Technol, 5 (1), 3.
- Harrington, H. J. (2000). Simulation modeling methods. To Reduce Risks and Increase Performance. McGraw-Hill Professional, 379.
- Mainframe computer – Wikipedia, the free encyclopedia [Elektronnyj resurs]. Rezhim dostupa: http://en.wikipedia.org/wiki/Mainframe_computer. Zagl. s ekrana.
- Mikihiko, O. (1998). Simulation engineering. IOS Press Amsterdam, 190.
- Fishman, G. S. (2001). Discrete-event simulation: modeling, programming, and analysis. Springer, 537.
- Decision Making Tools and Techniques – ProModel [Elektronnyj resurs]. Rezhim dostupa: <http://www.promodel.com/>. Zagl. s ekrana.
- Arena Portal: Products [Elektronnyj resurs]. Rezhim dostupa: http://www.arenasimulation.com/Products_Products.aspx. Zagl. s ekrana.
- Isee systems – Ithink Modeling & Simulation Software for Business [Elektronnyj resurs]. Rezhim dostupa: <http://www.iseesystems.com/software/Business/ithinkSoftware.aspx>. Zagl. s ekrana.
- Garrido, J. M. (2009). Object Oriented Simulation: A Modeling and Programming Perspective. Springer, 449.
- Heermann, D. W. (1990). Computer Simulation Methods in Theoretical Physics. Springer-Verlag Berlin Heidelberg, 155.
- Hartmann, S. (1996). The World as a Process: Simulations in the Natural and Social Sciences. Modelling and Simulation in the Social Sciences from the Philosophy of Science Point of View, Theory and Decision Library. Dordrecht: Kluwer, 77–100.
- Introduction to Military Training Simulation: A Guide for Discrete Event Simulationists [Elektronnyj resurs]. Rezhim dostupa: <http://www.modelbenders.com/papers/ml4des.html>. Zagl. s ekrana.
- Miller, S. (2000). Introduction to manufacturing simulation/ Scott Miller, Dennis Pegden. Proceedings of the 2000 Winter Simulation Conference.
- Banks, J. (1998). Handbook of simulation: principles, methodology, advances, applications, and practice, vol. 57. Wiley-IEEE, 849.
- Tomashevskij, V., Zhdanova, E. (2003). Imitacionnoe modelirovanie v srede GPSS, 416.
- Graudina, V., Grundspenik, J. (2005). Technologies and multi-agent system architectures for transportation and logistics support Vita Graudina. International Conference on Computer Systems and Technologies - CompSysTech, Varna, Bulgaria.
- Voroncov, K. V., Pshenichnikov, S. B. (2002). Imitacionnoe modelirovanie trgov: novaya texnologiya birzhevyyx trenazherov. Indikator, 2 (42).
Rezhim dostupa: http://www.forecsys.com/ru/site/about/media_coverage/exchange_simulator/. – Zagolovok s ekrana
- Shannon, R. (1978). Imitacionnoe modelirovanie sistem: iskusstvo i nauka. Moscow: Mir,
- Karpov, Yu. (2005). Imitacionnoe modelirovanie sistem. Vvedenie v modelirovanie s AnyLogic 5. SPb.: BXV-Peterburg, 400.
- Ljung, L., Glad, T. (1994). Modeling of dynamic systems. Hall PTR, 368.
- Borshhyov, A. A. Ot sistemnoj dinamiki i tradiconnogo IM – k prakticheskim agentnym modelyam: prichiny, texnologiya, instrumenty [Elektronnyj resurs]. Rezhim dostupa: <http://www.gpss.ru/>
- Forrester, J. W. (1968). Principles of systems. Wright-Allen Press Inc., 387.
- Forrester, J. (1958). Industrial Dynamics – A Major Breakthrough for Decision Makers. Harvard Business Review, 36, 4, 37–66.
- Richmond, B. (1994). System Dynamics/Systems Thinking: Let's Just Get On With It: International Systems Dynamics Conference in Sterling, Scotland. Springer, 215.
- Ruth, M., Hannon, B. (2003). Dynamic modeling for business management: an introduction. Springer, 388.

27. Jerry Banks, John S. Carson, Barry L. Nelson, David M. Nicol (2009). Discrete-Event System Simulation. Prentice Hall, 622.
28. Styuart, R. (2006). Iskusstvennyj intellekt: sovremennyy podhod. Moscow: Izdatelskij dom «Vilyams», 1408.
29. Yoav Shoham, Kevin Leyton-Brown (2008). Multiagent systems: algorithmic, game-theoretic, and logical foundations. Cambridge University Press, 483.
30. Industry Standards [Elektronnyj resurs]. Rezhim dostupa: <http://www.mak.com/community/standards.php>. Zagl. s ekrana.
31. Frederick Kuhl, Richard Weatherly, Judith Dahmann (1999). Creating Computer Simulation Systems: An Introduction to the High Level Architecture. Prentice Hall PTR, 224.
32. DIANE Publishing Company. Distributed Interactive Simulation of Combat: DIANE Publishing Company. – DIANE Publishing, 1996, 177.
33. CIGI: Overview [Elektronnyj resurs]. Rezhim dostupa: <http://cigi.sourceforge.net/overview.php>. Zagl. s ekrana.
34. The Foundation for Intelligent Physical Agents [Elektronnyj resurs]. Rezhim dostupa: <http://fipa.org/>. Zagl. s ekrana.
35. Subrahmanian, V. S. (2000). Heterogeneous agent systems. MIT Press, 580.
36. Standard Status Specifications [Elektronnyj resurs]. Rezhim dostupa: <http://fipa.org/repository/standardspecs.html>. Zagl. s ekrana.
37. Davydov, A. A. O Kompyutacionnoj teorii socialnyx agentov [Elektronnyj resurs]. Rezhim dostupa: http://www.isras.ru/files/File/Socis/1-6-2006/davydov_agents.pdf.
38. F. Zambonelli, N. Jennings, M. Wooldridge (2003). Developing MAS: The Gaia Methodology. ACM, 12, 3.
39. Object Management Group – UML [Elektronnyj resurs]. Rezhim dostupa: <http://uml.org/>. Zagl. s ekrana.
40. Agent UML [Elektronnyj resurs]. Rezhim dostupa: <http://www.auml.org/>. Zagl. s ekrana.
41. Juan C. Garcia-Ojeda, Jose de J. Perez-Alcazar, Alvaro E. Arenas (2004). Garcia-Ojeda. Extending the Gaia Methodology with Agent-UML. ACM, 3, 1456–1457.
42. A. Th. Schreiber, B. J. Wielinga, J. M. Akkermans, and W. Van de Velde (1994). CommonKADS: A comprehensive methodology for KBS development [Text] / A. Th. Schreiber, B. J. Wielinga, J. M. Akkermans, and W. Van de Velde // IEEE Expert, No 9(6).
43. The Prometheus Methodology [Elektronnyj resurs]. Rezhim dostupa: <http://www.cs.rmit.edu.au/agents/SAC2/methodology.html>. Zagl. s ekrana.;
44. MESSAGE - Methodology for Engineering Systems of Software Agents [Elektronnyj resurs]. Rezhim dostupa: <http://www.eurescom.de/public/projects/P900-series/p907/>. Zagl. s ekrana.
45. NetLogo [Elektronnyj resurs]. Rezhim dostupa: <http://ccl.northwestern.edu/netlogo/>. Zagl. s ekrana.
46. Understanding HubNet [Elektronnyj resurs]. Rezhim dostupa: <http://ccl.northwestern.edu/netlogo/hubnet.html>. Zagl. s ekrana.
47. What is NetLogo? [Elektronnyj resurs]. Rezhim dostupa: <http://ccl.northwestern.edu/netlogo/docs/>. Zagl. s ekrana.
48. VisualBots Software [Elektronnyj resurs]. Rezhim dostupa: <http://www.visualbots.com/software.htm>. Zagl. s ekrana.
49. MASON Multiagent Simulation Toolkit [Elektronnyj resurs]. Rezhim dostupa: <http://www.cs.gmu.edu/~eclab/projects/mason/>. Zagl. s ekrana.
50. Repast Agent Simulation Toolkit [Elektronnyj resurs]. Rezhim dostupa: <http://repast.sourceforge.net/>. Zagl. s ekrana.
51. Java [Elektronnyj resurs]. Rezhim dostupa: <http://java.com/>. Zagl. s ekrana.
52. Groovy – Documentation [Elektronnyj resurs]. Rezhim dostupa: <http://groovy.codehaus.org/Documentation>. Zagl. s ekrana.
53. Weka 3 – Data Mining with Open Source Machine Learning Software in Java [Elektronnyj resurs]. Rezhim dostupa: <http://www.cs.waikato.ac.nz/ml/weka/>. – Zagl. s ekrana.
54. Jade – Java Agent Development Framework [Elektronnyj resurs]. Rezhim dostupa: <http://jade.tilab.com/>. Zagl. s ekrana.
55. Kudryavcev, E. M. (2004). GPSS World. Osnovy imitacionnogo modelirovaniya razlichnyx sistem. Moscow: DMK Press, 320.
56. Gulyaev, A. (1999). MATLAB 5.2. Imitacionnoe modelirovanie v srede Windows. Moscow: Korona print, 288.
57. Okolnishnikov, V. V. (2005). Predstavlenie vremeni v imitacionnom modelirovanii. Vychislitelnye tekhnologii, 10, 5.
58. Ingo J. Timm, Dirk Pawlaszczyk (2005). Large Scale Multiagent Simulation on the Grid. IEEE International Symposium on Cluster Computing and the Grid (CCGRID), Cardiff University, Cardiff, UK.
59. OASIS SOA Reference Model TC [Elektronnyj resurs]. Rezhim dostupa: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=soa-rm. Zagl. s ekrana.
60. Axum [Elektronnyj resurs]. Rezhim dostupa: <http://msdn.microsoft.com/en-us/devlabs/dd795202.aspx>. Zagl. s ekrana.

Поступила (received) 05.03.2016

Бібліографічні описи / Библиографические описания / Bibliographic descriptions

Перспективы применения имитационного моделирования в задачах автоматизации и управления технологическими системами/ Р. Ю. Лопаткин, С. А. Петров, С. Н. Игнатенко, В. А. Иващенко // Вісник НТУ «ХПІ». Серія: Механіко-технологічні системи та комплекси. – Харків : НТУ «ХПІ», 2015. – No 17(1189). – С.61–71. – Бібліогр.: 60 назв. – ISSN 2079-5459.

Перспективы застосування імітаційного моделювання в задачах автоматизації та управлінні технологічними системами/ Р. Ю. Лопаткін, С. А. Петров, С. Н. Ігнатенко, В. А. Іващенко// Вісник НТУ «ХПІ». Серія: Механіко-технологічні системи та комплекси. – Харків : НТУ «ХПІ», 2015. – No 17(1189). – С.61–71. – Бібліогр.: 60 назв. – ISSN 2079-5459.

Prospects for the use of simulation modeling to automate tasks and the management of technological systems/ R. Y. Lopatkin, S. A. Petrov, S. N. Ignatenko, V. A. Ivaschenko//Bulletin of NTU “KhPI”. Series: Mechanical-technological systems and complexes. – Kharkov: NTU “KhPI”, 2015. – No 49 (1189).– P.61–71 – Bibliogr.: 60. – ISSN 2079-5459.

Відомості про авторів / Сведения об авторах / About the Authors

Лопаткін Роман Юрійович – кандидат фізико-математических наук, доцент кафедри загальної фізики, Інститут прикладної фізики НАН України, вул. Петропавлівська, 58, м. Суми, Україна, 40000, тел.: 050-572-39-61; e-mail: rlopatkin@gmail.com.

Лопаткин Роман Юрьевич – кандидат физико-математических наук, доцент кафедры общей физики, Институт прикладной физики НАН Украины, ул. Петропавловская, 58, г. Сумы, Украина, 40000, тел.: 050-572-39-61; e-mail: rlopatkin@gmail.com.

Lopatkin Roman – candidate of physical and mathematical sciences, associate professor of general physics department, Institute of Applied Physics, National Academy of Sciences of Ukraine, 58, Petropavlivska St., 40000 Sumy, Ukraine, тел.: 050-572-39-61; e-mail: rlopatkin@gmail.com.

Петров Сергій Олександрович – кандидат технічних наук, доцент кафедри комп'ютерних наук, Сумський державний університет, вул. Римського-Корсакова, 2, м. Суми, Україна, 41100, тел.: 066-726-32-29; e-mail: sergpet@gmail.com.

Петров Сергей Александрович – кандидат технических наук, доцент кафедры компьютерных наук, Сумской государственной университет, ул. Римского-Корсакова, 2, г. Сумы, Украина, 41100, тел.: 066-726-32-29; e-mail: sergpet@gmail.com.

Petrov Serhii – candidate of technical sciences, associate professor of computer science department, Sumy State University 2, Rymaskogo-Korsakova st., 40007 Sumy, Ukraine, phone.: 066-726-32-29; e-mail: sergpet@gmail.com.

Игнатенко Сергей Миколайович – старший науковий співробітник, Інститут прикладної фізики НАН України, вул. Петропавлівська, 58, м. Суми, Україна, 40000, тел.: 050-142-98-67; e-mail: mynameissergey@gmail.com.

Игнатенко Сергей Николаевич – старший научный сотрудник, Институт прикладной физики НАН Украины, ул. Петропавловская, 58, г. Сумы, Украина, 40000, тел.: 050-142-98-67; e-mail: mynameissergey@gmail.com.

Ignatenko Serhii – senior researcher, Institute of Applied Physics, National Academy of Sciences of Ukraine, 58, Petropavlivska St., 40000 Sumy, Ukraine, phone.: 050-142-98-67; e-mail: mynameissergey@gmail.com.

Іващенко Віталій Анатолійович – старший науковий співробітник, Інститут прикладної фізики НАН України, вул. Петропавлівська, 58, м. Суми, Україна, 40000, тел.: 095-925-77-00; e-mail: va.ivashchenko@gmail.com.

Ivashchenko Vitaliy – senior researcher, Institute of Applied Physics, National Academy of Sciences of Ukraine, 58, Petropavlivska St., 40000 Sumy, Ukraine, phone.: 095-925-77-00; e-mail: va.ivashchenko@gmail.com.