
**ЯЗЫКОВЫЕ И ПРОГРАММНЫЕ СРЕДСТВА ДЛЯ
МНОГОМОДЕЛЬНОГО ИССЛЕДОВАНИЯ ИМИТАЦИОННЫХ МОДЕЛЕЙ
КОМПЬЮТЕРНЫХ СЕТЕЙ****Е.Б. Замятина, Д.В. Мерзляков, А.А.Семеновых (Пермь)****Введение**

Для тщательного и качественного изучения сложных динамических систем и объектов, к которым можно отнести и компьютерные сети, целесообразно привлекать специалистов в различных областях знаний, имеющих опыт исследования с помощью того или иного математического аппарата. Так существует большое количество работ, посвященных исследованию компьютерных сетей, в основе которых лежит теория очередей [1,2,3,4] и т.д. При разработке алгоритмов маршрутизации целесообразно применять теорию графов (нахождение кратчайшего расстояния, определение характеристик графа, вершинами которого являются вычислительные узлы компьютерной сети). Для исследования трафика в компьютерных сетях[5], при моделировании атак в компьютерных сетях[6,7], для исследования функционирования параллельных и распределенных программных систем[8] используют теорию сетей Петри. Поэтому важно, чтобы при разработке имитационной модели исследователь смог оперировать привычными для него терминами, понятиями. В этом случае необходима настройка на конкретную предметную область, которую можно выполнить, используя языковой инструментарий, позволяющий разрабатывать предметно-ориентированные языки и осуществить перевод имитационной модели с одного визуального языка на другой (выполнить трансформацию имитационной модели) [9,10,11,12,13,14]. В результате можно осуществить многомодельное (полимодельное) исследование компьютерной сети [15]. Примером языкового инструментария, позволяющего выполнить трансформацию имитационной модели с одного визуального языка на другой, является MetaLanguage[10] - программная система, разработанная в ПГНИУ (Пермский государственный национальный исследовательский университет).

DSL-инструментарий

На первом этапе разработки DSL в языковом инструментарии MetaLanguage необходимо создать новую метамодель, указав её имя и описание (если это необходимо). Метамодель в данном случае – это предметно-ориентированный язык моделирования, который используется для создания моделей, ориентированных на решение конкретных задач. Затем можно приступить к построению метамодели с помощью графического редактора моделей.

При создании метамодели в первую очередь определяются базовые конструкции языка. Базовыми элементами, которые используются в MetaLanguage для создания метамodelей (DSL), являются сущность, отношение, ограничение. В процессе создания DSL определяются сущности метамодели, отношения между ними, задаются ограничения, налагаемые на сущности и отношения. После построения метамодели разработчик получает в распоряжение расширяемый, динамически настраиваемый визуальный язык моделирования.

Используя полученный DSL, исследователь создает модели. Разработанный язык может использоваться в качестве метаязыка для разработки новых языков. Трансформатор (часть программной системы Metalanguage) позволяет в соответствии с заданными правилами трансформаций (вертикальных и горизонтальных), созданными в

той же среде, преобразовать модели. Таким образом, разработанная модель может быть переведена на нужный язык и передана во внешние системы для решения соответствующих задач.

Авторы настоящей статьи предполагают создавать имитационные модели компьютерных сетей с использованием различных визуальных языков в системе автоматизированного проектирования и моделирования TRIADNS. В качестве метамodelей, необходимых для построения имитационной модели на том или ином визуальном языке, используют онтологии[16]. Рассмотрим программные средства системы моделирования компьютерных сетей TRIADNS.

Система имитационного моделирования Triad.Net и ее использование для моделирования компьютерных сетей (TRIADNS)

Система моделирования Triad.Net представляет собой совокупность лингвистических и программных средств имитационного моделирования[18].

СИМ Triad.Net включает следующие компоненты [19,20,21]: компилятор, ядро, графический редактор, подсистему отладки, подсистему валидации, подсистему синхронизации распределенных объектов модели, подсистему балансировки (распределенная версия), подсистему организации удаленного доступа, подсистему защиты от внешних и внутренних угроз, подсистему автоматического доопределения модели.

Назначение каждого из компонентов представлено ниже: TriadCompile (компилятор языка Triad, переводит описание имитационной модели с языка Triad во внутреннее представление); TriadDebugger (отладчик, использует механизм информационных процедур алгоритма исследования, локализует ошибки и вырабатывает рекомендации для их устранения на основании правил из базы данных, для каждого класса ошибок осуществляется поиск по онтологии соответствующего обработчика ошибок); TriadCore (ядро системы, включает библиотеки классов основных элементов модели), TriadEditor (редактор моделей, предназначен для работы с моделью как в удаленном, так и локальном режимах, локальный режим предполагает работу с системой в том случае, если нет удаленного доступа), TriadBalance[23] (подсистема балансировки), TriadSecurity (подсистема безопасности, этот компонент используют при удаленном доступе к системе моделирования), TriadBuilder[22] (подсистема автоматического доопределения частично описанной модели), база данных, где хранятся экземпляры элементов модели, TriadMining[24]-набор процедур для исследования результатов модели методами DataMining, TriadRule[23] – алгоритм синхронизации объектов распределенной модели, использующей для вычислительного эксперимента ресурсы нескольких вычислительных узлов.

Описание модели в системе Triad [18] можно определить как $M = (STR, ROUT, MES)$, где STR – слой структур, ROUT – слой рутин, MES – слой сообщений.

Слой структур представляет собой совокупность объектов, взаимодействующих друг с другом посредством посылки сообщений. Каждый объект имеет полюсы (входные и выходные), которые служат соответственно для приёма и передачи сообщений. Основа представления слоя структур – графы. В качестве вершин графа следует рассматривать отдельные объекты. Дуги графа определяют связи между объектами. Рутин описывают поведение объектов, они состоят из событий, планирующих друг друга. Одно из событий является входным, предназначено для обработки сообщений, которыми обмениваются объекты. Совокупность рутин определяет слой рутин ROUT.

Слой сообщений (MES) предназначен для описания сообщений сложной структуры.

Система Triad реализована таким образом, что пользователю необязательно описывать все слои. Так, если возникает необходимость в исследовании структурных особенностей модели, то можно описать только слой структур. В слое структур определены стандартные процедуры, с помощью которых можно определить множество вершин графа, множество ребер, дуг и т.д., выделить фрагмент модели для дальнейшего исследования, найти кратчайшее расстояние между двумя вершинами, компоненты связности $GetStronglyConnectedComponents(G)$, выделить слой структур из модели $GetGraphWithoutRoutes(M)$ и т.д.. TRIADNS располагает как текстовым так и графическим редакторами.

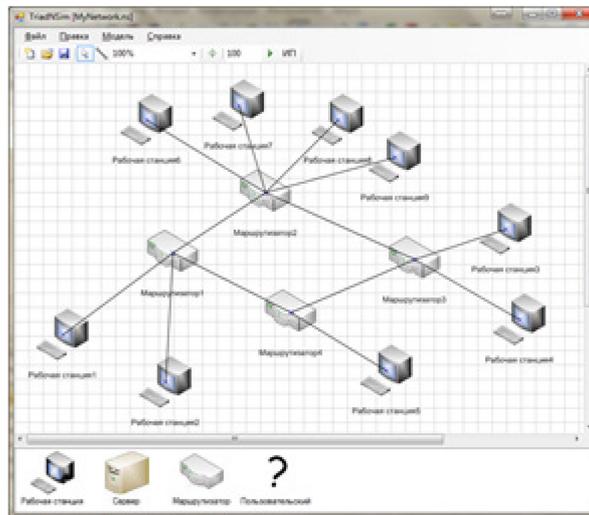


Рис. 1. Окно графического редактора

Следует обратить внимание, что в TRIADNS модель рассматривается как переменная. Она может быть построена с помощью операций над моделью. При построении модели используют графовые константы, которые соответствуют основным топологиям компьютерных сетей. Например, $dstar$ («звезда»), $dcycle$ («ориентированный цикл») и т.д.. Семантические типы (например, $Type Router$, $Host$) (типы «маршрутизатор», и «хост») применяют для того, чтобы можно было доопределить модель с помощью рутин, извлеченной из библиотеки экземпляров рутин.

Итак, *алгоритмом имитации* будем называть совокупность объектов, функционирующих по определенным сценариям, и синхронизирующий их алгоритм, а *алгоритмом исследования* – совокупность информационных процедур (процедуры для сбора статистических данных) и условий моделирования. В условиях моделирования перечисляют информационные процедуры для сбора статистики и критерии завершения имитационного эксперимента.

Информационные процедуры ведут наблюдение за элементами модели (событиями, переменными, входными и выходными полюсами), указанными пользователем. Если в какой-либо момент времени имитационного эксперимента пользователь решит, что следует установить наблюдение за другими элементами или выполнять иную обработку собираемой информации, он может сделать соответствующие указания, подключив к модели другой набор информационных процедур. В системе Triad.Net для анализа функционирования компьютерной сети

можно использовать стандартные и пользовательские информационные процедуры. Пользовательские информационные процедуры описывают на языке Triad. Для каждого элемента сети можно указать список необходимых информационных процедур, которые будут вести наблюдение во время моделирования за переменными, событиями и полюсами элемента. Система также предоставляет лингвистические средства для создания собственных условий моделирования, в которых можно описывать оригинальные алгоритмы сбора статистики и алгоритмы преобразования модели в динамике.

Онтологии и их применение в имитационном моделировании

Известно, что онтология – это описание типов сущностей предметной области, их свойств и отношений. Каждая предметная область (некая часть реального мира) может быть описана с помощью онтологий. Онтологии создаются и используются во множестве областей знаний, в том числе, известны примеры их успешного применения в имитационном моделировании. Однако создание онтологий для моделирования является достаточно сложной задачей, поскольку этот метод используют для исследования самых разнообразных систем, относящихся к различным предметным областям (химическим, физическим, транспортным и т.д.). Кроме того, методы имитационного моделирования основаны на математических, вероятностных и статистических расчетах, и, таким образом, онтологии для этих областей должны служить основой для всех остальных. Онтологии используют на различных этапах имитационного моделирования, начиная с этапа сбора информации о моделируемой системе и заканчивая этапом валидации модели [25]. Примерами использования онтологий моделирования могут служить управляемые онтологиями среды моделирования, а также подходы к объединению различных федератов, разрабатываемые для High Level Architecture (HLA) [26]. Подход, разрабатываемый для HLA, использует онтологии для описания требований, которым должны удовлетворять интерфейсы федератов для успешного взаимодействия в федерации, а так же для разработки этих требований с учетом знаний о моделируемой предметной области. В работе [27] представлена онтология портов, рассматриваемая как средство автоматизации построения моделей из компонентов. Порты описывают интерфейс, определяющий границы компонентов или подсистем в конфигурации системы. Система представлена как конфигурация подсистем или компонентов, соединенных друг с другом через четко определенные интерфейсы. Онтологии успешно применяются и в других работах по имитационному моделированию [28].

Итак, перед авторами стоит задача выполнить настройку системы имитации на конкретную предметную область с целью проведения качественных исследований и выполнить трансформации, например, преобразовать фрагмент модели компьютерной сети (рис.2.) в модель, описанную в терминах сети Петри (рис.3.).

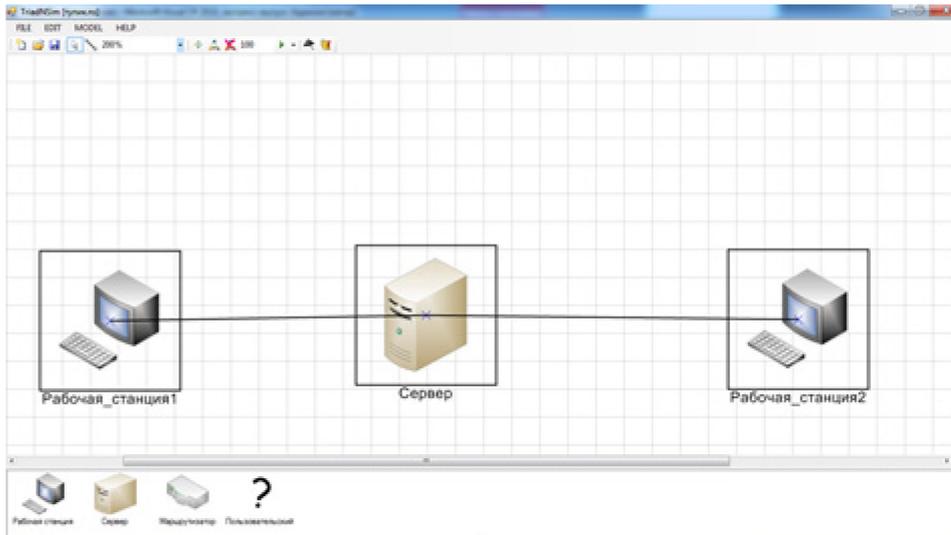


Рис.2. Пример компьютерной сети, состоящей из 2 рабочих станций и сервера

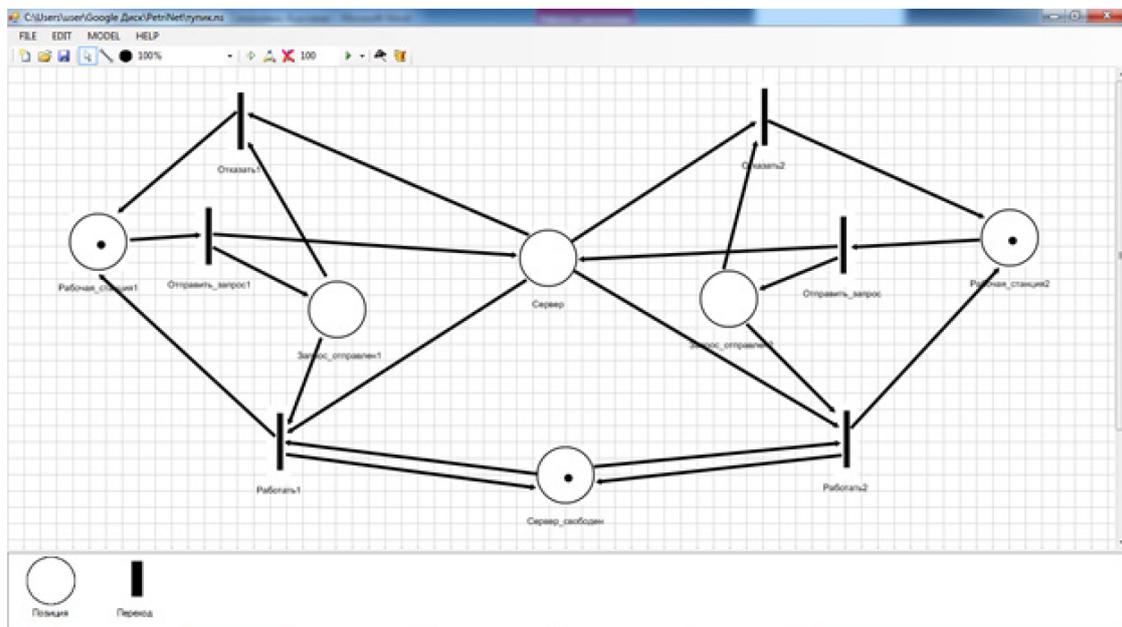


Рис.3. Фрагмент компьютерной сети, представленной на рис.2., в терминах сети Петри

Пусть каждая рабочая станция может занять ресурс, которым располагает сервер, и пусть некоторая рабочая станция занимает сервер на неопределенно большое время, и другие рабочие станции не смогут получить доступ к серверу. Используя встроенный DSL инструментарий для трансформации модели, получим модель в терминах сети Петри. Полученная модель позволяет выполнить анализ тупиковых ситуаций и блокировок. При выполнении трансформаций были использованы соответствующие правила преобразования и дополнительные структуры данных (календарь событий).

Для настройки на предметную область в качестве метамодели используют онтологии: базовая (мета метамодель, рис.4.), онтология для построения компьютерных сетей, онтология для описания модели в терминах сети Петри (рис.5.). В настоящее время в TRIADNS разработана также онтология для описания модели в виде системы

массового обслуживания (рис.6.). Для настройки на другую предметную область необходимо дополнить базовую онтологию новыми классами.

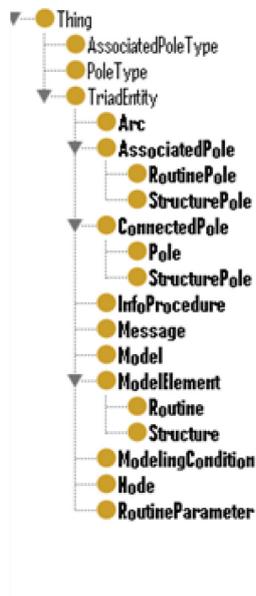


Рис.4. Базовая онтология

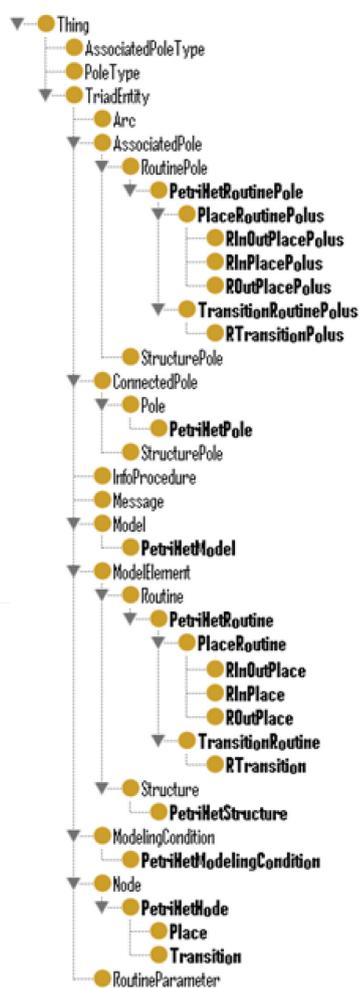


Рис.5.Онтология для сетей Петри

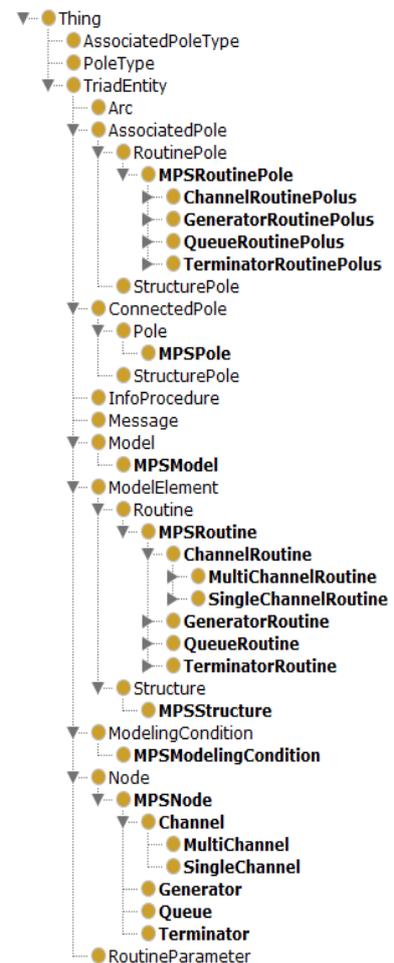


Рис.6. Онтология для систем массового обслуживания

Выводы

В статье представлен симулятор компьютерных сетей, который отличается от других программных средств моделирования компьютерных сетей некоторыми программными механизмами, позволяющими придать ему черты гибкого и эффективного программного средства. К этим средствам можно отнести возможность настройки имитационных моделей на конкретную предметную область и трансформацию их или фрагментов этих моделей в модель, в основе которой лежит иная математическая схема. Таким образом, для исследования можно применить иной математический аппарат, привлечь специалистов, владеющих этим аппаратом. Программные средства для трансформации моделей существуют (MetaLanguage), но в данном случае, благодаря встроенному в систему DSL-инструментария, пользователь не должен овладевать какими-то новыми программными средствами, он продолжает работать в привычной программной среде.

Благодарности

Работа выполнена при поддержке РФФИ (проект № 13-07-96506-р-юг-а)

Литература

1. Бахарева Н.Ф. Моделирование трафика в компьютерных сетях с помощью потоков событий / Н.Ф.Бахарева // Известия ВУЗов — Приборостроение. 2010, - Том 53, №12.- С. 13-22.
2. Бахарева Н.Ф. Анализ временных , характеристик непуассоновского трафика/ Н.Ф.Бахарева, И.В.Сарташевский //Электросвязь. 2010: №11. - С. 26-28.
3. Меркулова И.А. Расчет объема буфера для неотправленных данных в сетях Ethernet / И.А. Меркулова // V Междунар. конф. Молодых ученых и студентов «Актуальные проблемы современной науки»: тр. конф. -Самара, 2004. С. 30-32.
4. Ивановская (Синякова) И.А. Исследование модели параллельного обслуживания сдвоенных заявок в нестационарном режиме / С.П. Моисеева, И.А. Ивановская (Синякова) // Вестник Томского государственного университета. Управление, вычислительная техника и информатика. -2010. - № 3 (12). - С. 21 – 28
5. Гудов А. М., Семехина М. В. Имитационное моделирование процессов передачи трафика в вычислительных сетях // Управление большими системами, вып.31. – с. 130-161
6. Котенко Д.И., Котенко И.В., Саенко И.Б. Методы и средства моделирования атак в больших компьютерных сетях: состояние проблемы // Труды СПИИРАН. 2012. Вып. 3(22). С.5-30.
7. Котенко И.В., Степашкин М.В., Богданов В.С. Модель атак для имитации действий злоумышленника в системе анализа защищенности компьютерных сетей // IV Межрегиональная конференция "Информационная безопасность регионов России" ("ИБРР-2005"). Труды конференции. Санкт-Петербург. 2005.
8. Ломазова И.А. Вложенные сети Петри: моделирование и анализ распределенных систем с объектной структурой. М.: Научный мир, 2004.-208с.
9. Mikov A.I., Zamyatina E.B., Mikheev R.A. Towards the Flexibility of Software for Computer Network Simulation/ Mikov A.I //Proceedings of the 18th International Conference on Computers (part of CSCC '14) Advances in information science and applications - volumes I & II, Santorini Island, Greece, July 17-21, 2014, Edited by Prof. Nikos Mastorakis, ISBN: 978-1-61804-237-8, pp.391-397.
10. Лядова Л.Н. Многоуровневые модели и языки DSL как основа создания интеллектуальных CASE-систем /Л.Н. Лядова // Труды международных научно-технических конференций «Интеллектуальные системы» (AIS'08) и «Интеллектуальные САПР» (CAD-2008). Научное издание в 4-х томах. Т. 2. – М.: Физматлит, 2008. С. 37-41.
11. Лядова Л.Н. Подходы к описанию вертикальных и горизонтальных трансформаций метамоделей / Л.Н. Лядова, А.П. Серый, А.О. Сухов // Математика программных систем: межвуз. сб. науч. ст. – Пермь: Изд-во Перм. гос. нац. исслед. ун-та, 2012. – Вып. 9. – С. 33-49.
12. Лядова Л.Н. Визуальные языки и языковые инструментари: методы и средства реализации / Л.Н. Лядова, А.О. Сухов // Труды межд. научно-технической

- конференции «Интеллектуальные системы» (AIS'10). – М.: Физматлит, 2010. – Т. 1. – С. 374-382.
13. Сухов А.О. Инструментальные средства создания визуальных предметно-ориентированных языков моделирования / А.О. Сухов // *Фундаментальные исследования*. – 2013. – № 4 (ч. 4). – С. 848-852.
 14. Сухов А.О. Интеграция систем имитационного моделирования и предметно-ориентированных языков описания бизнес-процессов / А.О. Сухов // *Математика программных систем: межвуз. сб. науч. ст.* / Перм. гос. ун-т. – Пермь, 2009. С. 12-23.
 15. Соколов Б.В., Юсупов Р.М. Концептуальные основы квалиметрии моделей и полимодельных комплексов // *Имитационное моделирование. Теория и практика: сб. докл. 2-й всерос. науч.-практ. конф. ИММОД–2005*. Т. 1. СПб: ЦНИИТС. 2005. С. 65–70.
 16. Замятина Е.Б., Миков А.И., Михеев Р.А. Лингвистические и интеллектуальные инструментальные средства симулятора компьютерных сетей TRIADNS. *International Journal “Information theories & Applications (IJ ITA)*. Vol 19, Number 4, 2012, pp.355-368. ITHEA, Sofia, 1000, P.O.B. 775, Bulgaria. ISSN 1310-0513 (printed), ISSN 1313-0463 (online), ISSN 1313-0496 (CD/DVD)
 17. Cetinkaya D., Verbraeck A. Metamodeling and model transformations in modeling and simulation // *Proceedings of the 2011 Winter Simulation Conference*, 2011, pp. 3048-3058
 18. Mikov A.I. Formal Method for Design of Dynamic Objects and Its Implementation in CAD Systems // Gero J.S. and F.Sudweeks F.(eds), *Advances in Formal Design Methods for CAD, Preprints of the IFIP WG 5.2 Workshop on Formal Design Methods for Computer-Aided Design*, Mexico, Mexico, 1995. pp.105-127.
 19. Замятина Е.Б. Лингвистические и интеллектуальные инструментальные средства симулятора компьютерных сетей TRIADNS / Замятина Е.Б., Миков А.И., Михеев Р.А. // *International Journal “Information theories & Applications (IJ ITA)*. Vol 19, Number 4, 2012, pp.355-368. ITHEA, Sofia, 1000, P.O.B. 775, Bulgaria
 20. Mikov A., Zamiatina E. Program Tools and Language for Networks Simulation and Analysis. /Mikov A.//*Proceedings SDN & NFV – The Next Generation of Computational Infrastructure: 2014 International Science and Technology Conference «Modern Networking Technologies (MoNeTec)»* October 27-29, 2014 Lomonosov Moscow State University pp. 94-102
 21. Alexander I. Mikov, Elena B. Zamyatina, and Roman A. Mikheev. Towards the Flexibility of Software for Computer Network Simulation/ *Proceedings of the 18th International Conference on Computers (part of CSCC '14) Advances in information science and applications - volumes I & II*, Santorini Island, Greece, July 17-21, 2014, Edited by Prof. Nikos Mastorakis, ISBN: 978-1-61804-237-8, pp.391-397
 22. Mikov A., Zamyatina E., Kubrak E. Implementation of simulation process under incomplete knowledge using domain ontology //*Proceedings of the 6-th EUROSIM*

- Congress on Modeling and Simulation. University of Ljubljana, Slovenia, 2007, Vol.2. Book of full papers. p.1-7.
23. Mikov A., Zamyatina E., Kozlov A., Ermakov S. Some Problems of the Simulation Model Efficiency and Flexibility. Proceedings of «2013 8th EUROSIM Congress on Modelling and Simulation EUROSIM 2013», Cardiff, Wales, United Kingdom. P. 532-538.
 24. Kolevatov G.A., Zamyatina E.B. Simulation Analysis Framework Based on Triad.Net. Proceedings of the 6-th Spring/Summer Young Reseachers' Colloquium on Software Engineering. SYRCoSE 2012, Perm, May 30-31, 2012-Perm, Russia,pp.160-163.
 25. Sargent R.G. Some Recent Advances In The Process World View. // Proceedings of the 2005 Winter Simulation Conference / M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.,– pp. 293-298.
 26. Rathnam T., Paredis C.J.J. Developung Federation Object Models Using Ontologies // Proceedings of the 2004 Winter Simulation Conference / R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.,– pp. 1054-1062.
 27. Liang V.-C, Pardis C.J.J. A Port Ontology for Automated Model Composition // Proceedings of the 2003 Winter Simulation Conference / S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, eds., - pp. 613-622.
 28. Benjamin P., Akella K.V., Malek K., Fernandes R. An Ontology-Driven Framework for Process-Oriented Applications // Proceedings of the 2005 Winter Simulation Conference / M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.,– pp. 2355-2363.