

УДК 621.396.41

**Т.В. Ганджа, В.М. Дмитриев, Т.Н. Зайченко,
А.С. Букреев, А.М. Малышенко, А.А. Шилин**

Архитектура многопользовательской среды моделирования

Приводится архитектура клиент-серверной версии среды моделирования МАРС для реализации многопользовательских компьютерных тренажеров, позволяющих некоторым пользователям взаимодействовать с единой компьютерной моделью посредством локальных и глобальных вычислительных сетей. Описана методика сетевого взаимодействия клиент-серверной и структурно-функциональной схемы среды моделирования.

Ключевые слова: моделирование, автоматизация, компьютерное моделирование, клиент-сервер, удаленное выполнение эксперимента.

doi: 10.21293/1818-0442-2019-22-4-44-49

В настоящее время широко распространены программные инструменты для построения и анализа различных моделей, начиная от моделирования простых электрических цепей и заканчивая сложными имитационными моделями бизнес-процессов промышленных предприятий. Научным коллективом под управлением В.М. Дмитриева была разработана и развивается среда моделирования МАРС, основанная на методе *многоуровневого компьютерного моделирования* [1]. Он позволяет строить модели на трех уровнях: визуальном, логическом и объектном. Пользователь системы разрабатывает модель, вносит корректировки, запускает процесс расчета, анализирует выходные данные и графики, в общем итоге оформляет результаты моделирования в виде документов. Однако у данной системы есть два существенных недостатка. Во-первых, среда многоуровневого компьютерного моделирования МАРС [2, 3] может быть установлена на ПК без возможности сетевого взаимодействия. Весь процесс моделирования происходит локально, все результаты возможно только сохранить в файл. Во-вторых, однопользовательский режим работы, не допускающий одновременного взаимодействия нескольких пользователей с одной моделью, что актуально для построения сетевых компьютерных тренажеров [4].

В связи с этим возникает задача поиска или разработки инструмента для *сетевого многопользовательского компьютерного моделирования и анализа*, который должен отвечать следующим характеристикам:

1. Расчет моделей в сетевом вычислительном кластере (облаке).
2. Наличие удобного web-интерфейса для работы с моделями и их конфигурирования.
3. Многопользовательский доступ к процессу моделирования и анализу вычислений.
4. Динамическое изменение параметров модели при расчетах.
5. Параллельный запуск вычислений моделей.
6. Программный интерфейс взаимодействия (API) для различных прикладных тренажеров и приложений.
7. Просмотр результатов измерений в реальном времени.

После постановки задачи и определения требований к программному обеспечению был проведен сравнительный анализ среди инструментов компьютерного и имитационного моделирования. Рассматривались системы, которые либо имеют веб-приложение для разработки и анализа моделей, либо работают как тонкие клиенты (клиент-серверные приложения) и производят вычисления в удаленном сетевом кластере. Для сравнения были взяты современные системы моделирования: Simulink [5], Aris Cloud [6], AnyLogic [7], Simio Portal [8] и SimInTech [9].

Функциональные возможности системы Simulink были исследованы в [10]. Также, локальные версии систем моделирования Aris и Simulink рассматриваются в [11].

Данные источники описывают общие характеристики и функциональные возможности процесса моделирования. Однако для исследования возможности их использования в режиме многопользовательского сетевого моделирования и анализа был проведен сравнительный анализ на соответствие заданным требованиям. Результаты представлены в таблице.

Платформа AnyLogic обладает качественным и удобным веб-приложением, является SaaS (Software-as-a-Service) [12], но не поддерживает параллельный запуск моделей для расчетов и API. Simulink, в свою очередь, работает как тонкий клиент [13] без многопользовательского доступа и предлагает пользователям конфигурировать собственный вычислительный кластер в AWS (Amazon Web Services). Simio Portal и Aris Cloud не позволяют менять конфигурацию моделей и вести параллельные вычисления. Также ни одна из систем не поддерживает бесплатное API для получения и обработки результатов анализа.

Сравнительный анализ показал, что ни одна из исследуемых систем не может полностью отвечать задачам современного подхода к коллективному моделированию и анализу систем. Разработка и внедрение сетевой многопользовательской версии СМ МАРС направлена на повышение эффективности процесса моделирования и реализации многопользовательских компьютерных тренажеров, а также добавит множество новых функциональных преимуществ, таких как дистанционное многопользова-

тельское управление и наблюдение за экспериментом [14], возможность разработки различных клиентских приложений на разных платформах (мобильные приложения, индивидуальные 3D-тренажеры и т.п.). В дальнейшем, развивая идею среды моделирования с сетевым взаимодействием и добавляя в нее различный функционал коммуникативного взаимодействия, такой как видеоконференции и си-

стемы мгновенного обмена сообщениями в режиме реального времени, появится возможность разработать полноценный *стол технических совещаний* [15]. Он будет представлять собой систему, включающую в себя все этапы групповой работы над процессами создания и анализа компьютерных моделей технических объектов (ТО).

Сравнительный анализ систем компьютерного моделирования

Критерии \ Продукт	AnyLogic	Simulink	Simio Portal	Aris Cloud	SimInTech	СМ «МАРС»
Модель предоставления программного обеспечения	SaaS	Тонкий клиент	SaaS	SaaS	Тонкий клиент	SaaS
Наличие веб-интерфейса	Да	Нет	Да	Да	Нет	Да
Многопользовательский доступ к системе	Да	Нет	Да	Да	Да	Да
Конфигурирование моделей перед расчетами	Да	Да	Нет	Нет	Да	Да
Параллельный запуск вычислений модели	Частично (отдельный процесс браузера)	Да	Нет	Нет	Нет	Да
Наличие REST API	Нет	Нет	Нет	Платный функционал	Нет	Да
Вычислительный кластер	Собственные серверы	Amazon Web Services	Microsoft Azure	Собственные серверы	Собственный сервер	Собственные серверы
Интерфейс запуска вычисления модели	Графический	Программный	Графический	Графический	Графический	Графический
Просмотр результатов вычислений	В реальном времени	Да	Да	Выгрузка отчета.	В реальном времени	В реальном времени. Выгрузка отчета

Алгоритм функционирования протокола сетевого взаимодействия

Представленный на рис. 1 алгоритм функционирования протокола сетевого взаимодействия клиентов моделирующей системы позволяет нескольким пользователям на удаленных машинах осуществлять работу с одной компьютерной моделью ТО и наблюдать результаты ее анализа. Для этих целей он включает в себя следующие шаги:

Шаг 1. Пользователь подключается к системе и происходит выбор схемы (модели) из базы данных. В зависимости от выбора он может создать новую схему и начать процесс моделирования или выбрать сохраненную схему.

Шаг 2. Пользователь выбирает новую схему и задает параметры. Начинается процесс формирования новой модели по заданным параметрам пользователя. Созданная сущность сохраняется в базу данных. Когда схема сформирована и сохранена, происходит создание файла конфигурации. Он необходим для того, чтобы стандартизировать информацию о модели между различными узлами системы. Для реализации описания выбран язык разметки XML, поскольку он поддерживается практически всеми платформами и имеет стандартные библиотеки классов для чтения и записи файлов. Для сформированной схемы запускается отдельный процесс вычислительного ядра СМ МАРС [15], который получает на вход файл с конфигурацией схемы. Он непосредственно отвечает за построение схемы и все производимые расчеты. Весь процесс работы схемы:

внесение изменений в структуру модели, модификация параметров компонентов – происходит в режиме реального времени и называется *активной сессией*.

Система реагирует на любые изменения в активной сессии и в случае их появления инициирует процесс сохранения состояния схемы в базу данных.

Шаг 3. Когда пользователь выбирает уже сохраненную схему, структура и все параметры выгружаются из базы данных. На этом этапе система проверяет, нет ли уже запущенной сессии, работающей с данной схемой.

Шаг 4. Выполняется в случае, если схема запущена (например, другой пользователь начал процесс моделирования раньше). Система подключается к текущему процессу анализа модели. При этом пользователь в реальном времени начинает наблюдать результаты работы активной сессии.

Шаг 5. Выполняется в случае, когда для выбранной модели нет активной сессии. Все ее параметры загружаются из базы данных, после чего начинается процесс генерации файла конфигурации. Запускается отдельный процесс вычислительного ядра СМ МАРС, в который передается файл конфигурации.

Шаг 6. Является заключительным этапом. Каждая из предыдущих условных цепочек действий заканчивается переходом к этому шагу. Работающий процесс СМ МАРС в бесконечном цикле принимает параметры от пользователя по сети. На каждой итерации происходит анализ переданных пользовательских параметров и запускается расчет модели. Результат расчета передается обратно пользователю.

Эти действия происходят в режиме реального времени с задержкой t :

$$t = t_{ch} + t_r,$$

где t – общее время задержки (время перехода информации о результатах вычисления от процесса

СМ МАРС к клиентскому приложению пользователя); t_{ch} – время задержки канала (зависит от пропускной способности сетевого канала); t_r – время вычисления модели (прямо пропорционально сложности модели и мощности аппаратного обеспечения).

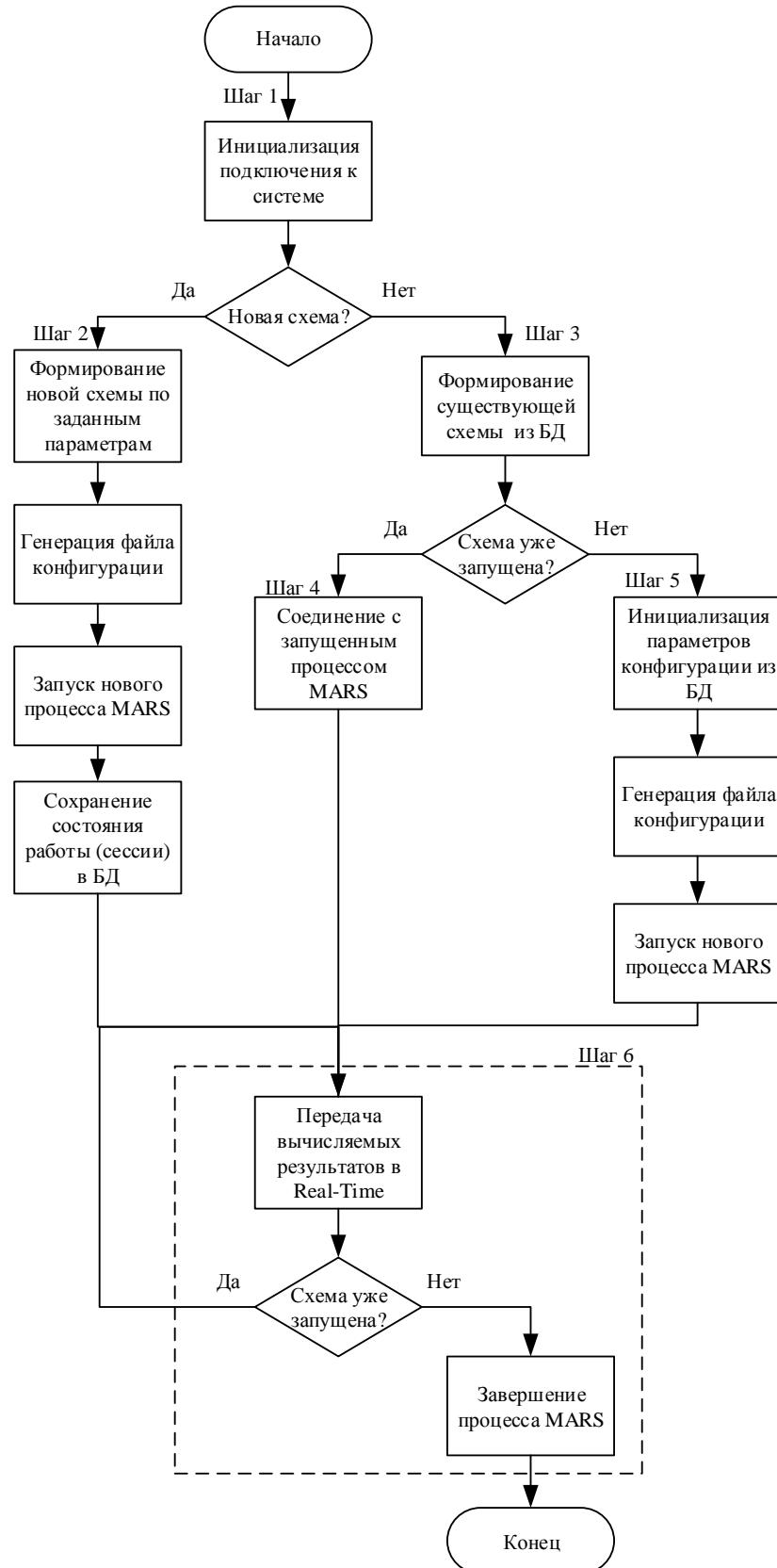


Рис. 1. Алгоритм функционирования протокола сетевого взаимодействия систем

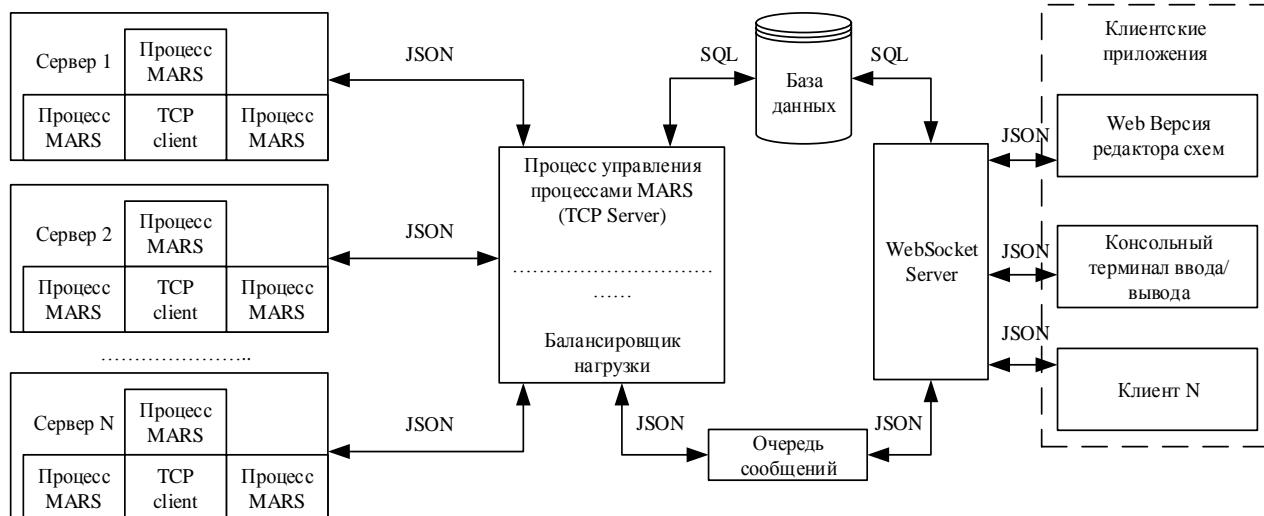


Рис. 2. Структурно-функциональная схема многопользовательской среды моделирования МАРС

Условием выхода из бесконечного цикла является завершение активной сессии по инициативе пользователя (либо в случае возникновения аварийной ситуации). В результате происходит завершение процесса СМ МАРС, после чего все данные сохраняются в базу данных.

Архитектура сетевой структуры СМ МАРС

На рис. 2 представлена функциональная схема сетевой структуры СМ МАРС, на которой изображена клиент-серверная архитектура системы [16].

Центральным звеном системы являются два компонента. Первый из них – WebSocket Server – представляет собой отдельное приложение, которое реализует протокол WebSocket. Он устанавливает полнодуплексное соединение с клиентскими приложениями поверх TCP-соединения и предназначается для обмена данными между браузером или другим клиентским приложением, которое реализует данный протокол, и веб-сервером. Главная задача сервера – обеспечение обмена данными с любыми клиентскими приложениями, которые поддерживают данный вид связи. WebSocket Server также реализует основной принцип функционирования сетевой архитектуры, содержит методы создания новой модели, установку произвольных значений параметров для ее компонентов, сохранение модели в базу данных. В качестве базы данных может использоваться любая современная реляционная БД с открытым исходным кодом. Обмен данными между клиентом и сервером происходит в формате JSON-пакетов. Для тестирования работы сервера разработан эмулятор – консольное приложение, которое будет организовывать простые операции ввода/вывода и обмениваться данными с WebSocket Server [17].

После того как запрос от клиента переходит к веб-серверу, он обрабатывается и направляется в очередь обмена сообщениями (Message Queue). Она необходима для создания канала, по которому будет происходить обмен данными между веб-сервером и сервисом управления процессами СМ МАРС. По умолчанию очередь сообщений имеет только одно-

стороннюю связь между клиентом и сервером, поэтому необходимо развернуть два канала для обмена данными в обе стороны. В качестве реализации очереди сообщений предполагается сервер RabbitMQ.

Сервис управления процессами СМ МАРС является вторым центральным звеном системы. Он является промежуточным сервисом между вычислительным ядром СМ МАРС и клиентским приложением. Частью сервиса также является балансировщик нагрузки – автоматизированная система управления пулом серверов. В момент, когда необходимо запустить анализ новой модели и начать вычислительный эксперимент, балансировщик в простом случае проверяет загруженность серверов другими процессами СМ МАРС и запускает новый процесс на свободном сервере. В более сложном случае реализация алгоритма распределения нагрузки представляет собой добавление специальных программных агентов, которые представляют собой TCP-клиенты и предназначены для сбора информации как о работе сервера, так и общей информации о процессе МАРС. В этом случае балансировщик выбирает сервер для запуска с учетом вектора этих параметров. В идеале загрузка серверов по всем параметрам должна быть равномерной.

В процессе работы вычислительный процесс отправляет данные к сервису управления процессами, далее в очередь сообщений, после чего они переходят в WebSocket Server, записываются в базу данных и попадают к клиентам.

Заключение

Предложенная структура сетевой многопользовательской среды моделирования открывает возможности реализации виртуальных столов технических совещаний, в которых несколько пользователей взаимодействуют с одной моделью, анализируемой на сервере. Это позволяет реализовать новую структуру сетевых компьютерных тренажеров, обеспечивающих совместное управление компьютерными моделями технических и технологических объектов, обучения на реальных прототипах которых связано с большими рисками для жизни и с существенными

экономическими затратами. Реализация такой структуры программного обеспечения стала возможной благодаря использованию многоуровневой структуры компьютерной модели и реализации на логическом уровне компонентов сетевого взаимодействия моделей.

Литература

1. Дмитриев В.М. Принцип формирования многоуровневых компьютерных моделей SCADA-систем для управления сложными технологическими объектами / В.М. Дмитриев, Т.В. Ганджа // Информатика и системы управления. – 2013. – № 2 (36). – С. 24–35.
 2. МАРС – среда моделирования технических устройств и систем / В.М. Дмитриев, А.В. Шутенков, Т.Н. Зайченко, Т.В. Ганджа. – Томск: В-Спектр, 2011. – 278 с.
 3. Дмитриев В.М. Среда многоуровневого компьютерного моделирования химико-технологических систем / В.М. Дмитриев, Т.В. Ганджа. – Томск: Изд-во Том. ун-та, 2017. – 332 с.
 4. Dmitriev V.M., Gandzha T.V., Dolganov I.M., Pisarev M.O., Dolganova I.O., Sizova E.N., Ivashkina E.N. Structure of network simulator for training and retraining of operators of controlled technological objects of oil and gas industry // Petroleum and Coal. – 2015. – Vol. 57, Is. 6. – P. 691–695.
 5. Дьяконов В.П. Simuink 5/ 6/ 7: Самоучитель. – М.: ДМК-Пресс, 2008. – 784 с.
 6. ARIS Cloud for Students | ARIS BPM Community [Электронный ресурс]. – Режим доступа: <https://www.aris-community.com/university/arис-cloud-students>, свободный. (дата обращения: 20.06.2019).
 7. Карпов Ю.Г. Имитационное моделирование систем. Введение в моделирование с AnyLogic 5. – СПб.: БХВ-Петербург, 2006. – 400 с.
 8. Simio Simulation Software Reviews; Overveiw; Pricing and Features [Электронный ресурс]. – Режим доступа: <https://reviews.financesonline.com/p/simio-simulation-software/> свободный (дата обращения: 20.06.2019).
 9. Среда динамического моделирования технических систем SimInTech / Б.А. Карташов, А.М. Щекатуров, Е.А. Шабаев, О.С. Козлов. – М.: ДМК-Пресс, 2017. – 424 с.
 10. Рванцов Ю.А. Сравнительный анализ систем имитационного моделирования деловых процессов по критерию функциональной полноты // Вестник ДГТУ. – 2011. – Т. 11, № 1(52). – С. 69–73.
 11. Бондаренко А.А. Сравнительный анализ систем имитационного моделирования для исследования технологических процессов обработки информации в глобально распределенных автоматизированных информационных системах // Software & Systems. Программные продукты и системы. – 2014. – №3 (107). – С. 47–52.
 12. Модель SaaS – в мире и России [Электронный ресурс]. – Режим доступа: <https://www.bytemag.ru/articles/detail.php?ID=12825>, свободный (дата обращения: 11.11.2019).
 13. Петухов Р.Н. Применение технологии «тонкий клиент» на промышленных предприятиях // Молодой учёный. – 2016. – № 17 (121). – С. 71–74.
 14. Naumovic M.B., Zivanovic D. Remote Experiments in Control Engineering Education Laboratory // Intern. Journ. of Online Engineering. – 2008. – Vol. 4, No. 2. – P. 48–53.
 15. Буданов А.Н. Методика организации стола технических совещаний / А.Н. Буданов, В.М. Дмитриев // Доклады ТУСУР. – 2014. – № 1 (31). – С. 201–204.
 16. Дмитриев В.М. Архитектура универсального вычислительного ядра для реализации виртуальных лабораторий / В.М. Дмитриев, А.В. Шутенков, Т.В. Ганджа // Приборы и системы. Управление. Контроль. Диагностика. – 2004. – № 2. – С. 24–28.
 17. Фаулер М. Архитектура корпоративных программных приложений. – М.: ИД «Вильямс», 2007. – 544 с.
 18. Len Bass, Paul C. Clements, Rick Kazman. Software architecture in practice. – Third edition. – Bengaluru: Pearson Education India, 2013. – 622 p.
-
- Ганджа Тарас Викторович**
Д-р техн. наук, профессор каф. компьютерных систем в управлении и проектировании (КСУП)
Томского государственного университета
систем управления и радиоэлектроники (ТУСУР)
Ленина пр-т, д. 40, г. Томск, Россия, 634050
Тел.: +7-913-846-11-77
Эл. почта: gandgatv@gmail.com

Дмитриев Вячеслав Михайлович
Д-р техн. наук, профессор каф. КСУП ТУСУРа
Ленина пр-т, д. 40, г. Томск, Россия, 634050
Тел.: +7 (382-2) 41-39-15
Эл. почта: dmitriewvm@gmail.com

Зайченко Татьяна Николаевна
Д-р техн. наук, профессор каф. КСУП ТУСУРа
Ленина пр-т, д. 40, г. Томск, Россия, 634050
Тел.: +7 (382-2) 41-39-15
Эл. почта: zrn@ie.tusur.ru

Букреев Александр Сергеевич
Аспирант каф. КСУП ТУСУРа
Ленина пр-т, д. 40, г. Томск, Россия, 634050
Тел.: +7 (382-2) 41-39-15
Эл. почта: alexander.eleventh@gmail.com

Малышенко Александр Максимович
Д-р техн. наук, профессор отделения автоматизации и робототехники Инженерной школы информационных технологий и робототехники (ИШИТР)
Национального исследовательского
Томского политехнического университета (НИ ТПУ)
Ленина пр-т, д. 30, г. Томск, Россия, 634050
Тел.: +7 (382-2) 701837
Эл. почта: mam@tpu.ru

Шилин Александр Анатольевич
Д-р техн. наук, профессор
Инженерной школы энергетики отд. электроэнергетики и электротехники (ОЭЭ ИШЭ) НИ ТПУ
Ленина пр-т, д. 30, г. Томск, Россия, 634050
ORCID: 0000-0002-4761-7249
Тел.: +7-903-954-00-88
Эл. почта: shilin@tpu.ru
- Gandzha T.V., Dmitriev I.V., Zaychenko T.N.,
Bukreev A.S., Malysheenko A.M., Shilin A.A.
Architecture of multiuser simulation environment MARS

The architecture of the client-server version of the simulation environment MARS for the implementation of multi-user computer simulators that allow several users to interact with a single computer model through local and global computer networks is presented. The methodology of the network-client-

server interaction and the structural-functional diagram of the modeling environment are described.

Keywords: modeling, automation, computer modeling, client-server, remote execution of the experiment.

doi: 10.21293/1818-0442-2019-22-4-44-49

References

1. Dmitriev V.M. Gandzha T.V. [The principle of formation of multi-level computer models of SCADA-systems for control complex technological objects]. *Computer Science and Control Systems*, 2013, no. 2 (36), p. 24–35 (in Russ.).

2. Dmitriev V.M., Shutenkova A.V., Zaichenko T.N., Gandzha T.V. *MARS – sreda modelirovaniya tekhnicheskikh ustroystv i sistem* [MARS – environment for modeling technical devices and systems]. Tomsk. In-Spectrum, 2011, 278 p. (in Russ.).

3. Dmitriev V.M., Gandzha T.V. *Sreda mnogourovnevogo kompyuternogo modelirovaniya khimiko-tehnologicheskikh system* [The environment of multi-level computer simulation of chemical-technological systems]. Tomsk, Izd-vo Tom, un-ta, 2017. 332 p. (in Russ.).

4. Dmitriev V.M., Gandzha T.V., Dolganov I.M., Pisarev M.O., Dolganova I.O., Sizova E.N., Ivashkina E.N. Structure of network simulator for training and retraining of operators of controlled technological objects of oil and gas industry. Petroleum and Coal. 2015, vol. 57, Is. 6, pp. 691–695.

5. Dyakonov D.P. *Simulink 5/ 6/ 7: Samouchitel'* [Simulink 5/ 6/ 7: tutorial]. Moscow, DMK-Press, 2008, 278 p. (in Russ.).

6. ARIS Cloud for Students. ARIS BPM Community. – Access mode: <https://www.ariscommunity.com/> universitiy/aris-cloud-students, free (accessed: June 20, 2019).

7. Karpov Y.G. *Imitatsionnoye modelirovaniye sistem. Vvedeniye v modelirovaniye s AnyLogic 5.* [Simulation systems. Introduction to modeling with AnyLogic 5]. SPb.: BHV:Peterburg, 2006, 400 p.

8. Simio Simulation Software Reviews; Overveiw; Pricing and Features. – Access mode: <https://reviews.financesonline.com/p/simio-simulation-software/>, free (accessed: June 20, 2019).

9. Kartashov B.A., Shchekaturov A.M., Shabayev Ye.A., Kozlov O.S. *Sreda dinamicheskogo modelirovaniya tekhnicheskikh sistem SimInTech* [SimInTech Technical Systems Dynamic Simulation Environment]. Moscow, DMK-Press, 2017, 424 p. (in Russ.).

10. Rvancov Y.A. Comparative analysis of business processes simulation systems by functional completeness criterion. *Vestnik of Don State Technical University*, 2011, vol. 11, no. 1 (52), pp. 69–73.

11. Bondarenko A.A. Simulation system comparative analysis for information processing research in a globally distributed automated information systems. *Software and System*, 2014, no. 3 (107), pp. 47–52.

12. Model' Saas – v mire i Rossii [Saas model – in the world and Russia]. Available at: <https://www.bytemag.ru/articles/detail.php?ID=12825>, (accessed: November 11, 2019).

13. Petuhov R.N. The use of thin client technology in industrial enterprises. *Young scientist*, 2016, no. 17 (121), pp. 71–74.

14. Naumovic M.B., Zivanovic D. Remote Experiments in Control Engineering Education Laboratory. Intern. Journ. of Online Engineering, 2008, vol. 4, no. 2, pp. 48–53.

15. Budanov A.N., Dmitriev V.M. Methods of organizing the table of technical meetings. *Proceedings of TUSUR University*, 2014, no 1 (31), pp. 201–204 (in Russ.).

16. Dmitriev V.M., Shutenkova A.V., Gandzha T.V. The architecture of a universal computing kernel for the implementation of virtual laboratories. *Instruments and systems. Management. Control, Diagnostic*, 2004, no. 2, pp. 24–28 (in Russ.).

17. Fauler M. *Arkhitektura korporativnykh programnykh prilozheniy* [Enterprise Software Architecture]. Moscow, Vilyams, 2007, 544 p.

18. Len Bass, Paul C. Clements, Rick Kazman. Software architecture in practice, third edition, Bengaluru. Pearson Education India, 2013, 622 p.

Taras V. Gandzha

Doctor of Engineering Sciences,
Department of Computer Control and Design Systems,
Tomsk State University of Control Systems
and Radioelectronics
40, Lenin pr., Tomsk, Russia, 634050
Phone: +7-913-846-11-77
Email: gandgatv@gmail.com

Vjacheslav M. Dmitriev

Doctor of Engineering Sciences, professor,
Department of Computer Control and Design Systems,
Tomsk State University of Control Systems
and Radioelectronics
40, Lenin pr., Tomsk, Russia, 634050
Phone: +7 (382-2) 41-39-15
Email: dmitriewvm@gmail.com

Tatyana N. Zaychenko

Doctor of Engineering Sciences,
Department of Computer Control and Design Systems,
Tomsk State University of Control Systems
and Radioelectronics
40, Lenin pr., Tomsk, Russia, 634050
Phone: +7 (382 2) 41-39-15
Email: zrn@ie.tusur.ru

Alexandr A. Bukreev

PhD student Department of Computer Control and Design Systems, Tomsk State University of Control Systems
and Radioelectronics
40, Lenin pr., Tomsk, Russia, 634050
Phone: +7-960-978 86-68
Email: alexander.eleventh@gmail.com

Aleksandr M. Malysheenko

Doctor of Engineering Sciences, Professor
Department of Automation and Robotics,
Engineering School of Information Technology and Robotics,
National Research Tomsk Polytechnic University
30, Lenin pr., Tomsk, Russia, 634050
Phone: +7 (382-2) 70-18-37
Email: mam@tpu.ru

Aleksander A. Shilin

Doctor of Engineering Sciences, Professor,
Power Engineering School, Department of Electric Power
and Electrical Engineering,
National Research Tomsk Polytechnic University
30, Lenin pr., Tomsk, Russia, 634050
ORCID: 0000-0002-4761-7249
Phone: +7-903-954-00-88
Email: shilin@tpu.ru