

## ЭКСПЕРТНЫЕ СИСТЕМЫ В КОМПЬЮТЕРНЫХ СЕТЯХ: ВОПРОСЫ СОЗДАНИЯ ПРИ WEB-РАЗРАБОТКАХ

Василенко К.А.<sup>1</sup>, Золкин А.Л.<sup>2</sup>, Абрамов Н.В.<sup>3</sup>, Курганов Д.О.<sup>3</sup>

<sup>1</sup> Владивостокский государственный институт экономики и сервиса, Владивосток, РФ

<sup>2</sup> Поволжский государственный университет телекоммуникаций и информатики, Самара, РФ

<sup>3</sup> Дальневосточный федеральный университет, Владивосток, РФ

E-mail: k2857@mail.ru, alzolkin@list.ru, nikolay.abramov1990@mail.ru, kurganov\_vl@mail.ru

Рассматриваются проблемы разработки экспертных систем в вычислительных сетях при web-программировании. Прослеживается зависимость особенностей обработки, хранения и передачи информации и web-приложений от уникальности и проблематики создания экспертных систем. При данных обстоятельствах имеются затруднения, связанные с загруженностью сетевого трафика, когда информация может быть подвергнута угрозам целостности, поскольку будет отсутствовать необходимая синхронизация данных между клиентскими службами и службами серверов. С целью предотвращения и ликвидации указанной проблемы предлагается надлежащим образом выбрать комплекс web-инструментов и приложений. Для достижения этой цели необходимо сделать выборку и проанализировать языки web-программирования, определить для создания экспертных систем соответствующий специальным требованиям язык. При этом следует учитывать возможности web-приложений для клиентской стороны при активно развивающихся информационных технологиях.

**Ключевые слова:** вычислительные сети, web-сервер, языки программирования, экспертные системы, web-разработки, информация, проектирование

### Введение

В настоящее время экспертные системы вышли на новый уровень наряду с квалифицированными специалистами в рамках анализа и решения сложных вопросов и задач, при этом они выдают весьма эффективные и качественные результаты. К экспертным системам можно отнести компьютерные системы, обладающие способностью частичной замены специалистов-экспертов при разрешении проблемных ситуаций [1].

### Проблема исследования экспертных систем вычислительных сетей

Экспертные системы все чаще находят свое применение в виде онлайн-систем с открытым доступом для различного контингента пользователей [2]. Одной из наиболее серьезных проблем при построении таких систем является сложность проектирования, так как необходимо асинхронное взаимодействие клиентской и серверной сторон. Сложность заключается в структурировании и извлечении, а также хранении информации, и прежде всего возникает вопрос о месте ее хранения. Экспертная система не хранит в себе конфиденциальных данных, но и размещать каждый раз данные на клиентскую часть будет неуместным ввиду большого объема информации.

Размещение базы на серверной стороне заставит клиентскую часть постоянно запрашивать данные с серверной стороны небольшими частями, а если учесть еще ряд дополнительной информации, посылаемой вместе с пакетом дан-

ных, то объем будет так же велик, как и хранение на клиентской стороне [3]. Следовательно, проблема хранения данных обусловлена высокой задержкой передачи информации по каналам связи.

Вопрос структурирования информации также должен быть регламентирован отношением клиентской и серверной сторон. Суть вопроса заключается в неудобном для проектирования и программирования формировании структуры, по которой данные будут укомплектовываться в систему и затем извлекаться.

Проблема состоит в непригодности существующих web-технологий к решению такого рода задач. Исходя из вышеописанного, задача разработки экспертных систем на базе web-технологий является значимой по причине стремительного развития ЭС в глобальной сети Internet [4].

Целью исследования является анализ метода решения установленной проблемы, основанный на комплектации специального пакета инструментов, состоящих из стандартных средств разработки в сфере web-технологий, которые имеют максимальную производительность, удобство и скорость разработки, а также некоторую специфику, которая дает возможность решения задачи нетрадиционными способами.

### Сравнительный анализ серверной технологии web-программирования

Для того чтобы выявить необходимые качества в средствах разработки, следует провести

сравнительный анализ серверной технологии web-программирования. Для этого возьмем следующие языки программирования: PHP; Perl; Python; Ruby и JavaScript.

Язык JavaScript классически принято считать клиентским и интерпретируемым строго в браузере клиента, но ввиду его некоторой специфики был разработан фреймворк Node.JS, представляющий собой программную платформу, основанную на движке V8 (транслирующем JavaScript в машинный код), которая превращает JavaScript из узкоспециализированного языка в язык общего назначения [5].

Node.JS предоставляет возможность JavaScript взаимодействовать с устройствами ввода/вывода через свои API (написанном на C++), подключать другие внешние библиотеки, написанные на разных языках, обеспечивая вызовы к ним из JavaScript-кода [6].

Node.JS применяется преимущественно на сервере, выполняя функцию веб-сервера, при этом есть возможность разрабатывать на Node.JS и десктопные оконные приложения. Node.JS производительнее других серверных языков.

Сравнительный анализ будет производиться по следующим свойствам: парадигмам; типизации; управлению памятью; управлению потоком вычислений; функциональным возможностям; компилятору/интерпретатору. Данные о языках web-программирования приведены в таблице.

По данным, представленным в таблице, можно сделать вывод, что по производительности разработчика все языки имеют разные параметры, однако, исходя из производительности работы и потребления вычислительных ресурсов, можно выявить один оптимальный язык программирования – JavaScript (Node.JS), который не уступает по производительности разработки другим языкам.

Еще одной из причин, по которой Node.JS считается наиболее актуальной технологией для выбора при разработке ЭС, является тот факт, что на серверной стороне будет тот же самый язык, что и на клиентской, что существенно повлияет на удобство написания кода и на взаимодействие клиента с сервером [7].

Также при использовании Node.JS серьезно упрощается процесс передачи данных. Это также можно назвать преимуществом серверного JavaScript, дающим возможность самостоятельно написать собственный web-сервер. При рассмотрении клиентской части следует обратить внимание на то, что в настоящее время ряд браузеров стремительно развивает новый стандарт

HTML5, который открывает перед разработчиком новые возможности работы с клиентской стороной, в частности организацию на стороне клиента базы данных, основанной на СУБД SQLite3 [8].

Такой подход дает возможность обеспечить логическое кеширование и сократить трафик между клиентом и сервером, тем самым увеличить скорость работы экспертной системы. Нужно помнить, что, говоря о серверных языках, пользователь должен запросить вход на web-страницу, куда будет произведен переход на указанную страницу по определенной ссылке имеющегося браузера [9]. Далее будет вестись обработка на сервере для выполнения программных приложений, имеющих связь со страницей. После указанных действий будет дан ответ пользователю вычислительной сети в виде файла, имеющего различные расширения, в том числе Perl, SI, XHTML, HTML и другие [10].

Работоспособность программных приложений уже будет иметь определенную зависимость от сервера, где будет находиться сам сайт, а также от версии языка программирования, которым он поддерживается. Особенностью функционирования серверных языков программирования является их способность организовывать контакты с системой управления базами данных (СУБД), сервером СУБД.

Язык программирования PHP имеет открытый исходный код и в ряду своих сценариев обладает способностью интеграции в условно обыкновенные html-документы. Однако при этом на web-сервере должен быть интерпретатор языка PHP [11]. Указанный язык может служить при написании системных программ или их составляющих. Написанные на данном языке web-приложения имеют сложности с безопасностью, при этом отсутствует возможность асинхронной и многопоточной работы.

Язык программирования Ruby характеризуется способностью к объектно-ориентированному функционалу, он независим от операционной системы в рамках многопоточности и управления оперативно запоминающим устройством. Однако данный язык отличается особой сложностью при его изучении, он весьма непродуман, мало развит.

Язык программирования Python нацелен на решение сложных вопросов при web-разработках. Данный язык генерирует комплекс статистических html-страниц, находящихся в обслуживании серверов. Python весьма ограничен в инструментарии для работы со СУБД, его

Таблица. Сравнение языков WEB-программирования по их различным свойствам

Наименование языка	Парадигмы	Типизация	Управление памятью	Управление потоком вычислений	Функциональные возможности	Компилятор / интерпретатор
JavaScript	Императивная, объектно-ориентированная, функциональная (частично), рефлексивная, обобщенное программирование, декларативная (частично)	Динамическая, неявная, неявное приведение типов без потери данных, неявное приведение типов в неоднозначных ситуациях	Сборка мусора (освобождения памяти в куче, занятой неиспользуемыми объектами)	Инструкции break без метки, инструкции break с меткой, поддержка try/ catch, блок finally	First class functions, анонимные функции, лексические замыкания, каррирование	Open-source компилятор (интерпретатор), возможность компиляции, Bootstrapping, интерпретатор командной строки, условия компиляции (ограниченно)
Perl	Императивная, объектно-ориентированная, функциональная, рефлексивная, обобщенное программирование декларативная (ограниченно)	Статическая (частично), динамическая, неявная, неявное приведение типов без потери данных, неявное приведение типов с потерей данных, неявное приведение типов в неоднозначных ситуациях, информация о типах в runtime, информация о типах-параметрах в runtime информация о типах в runtime (ограниченно), информация о типах-параметрах в runtime (ограниченно)	Сборка мусора (освобождения памяти в куче, занятой неиспользуемыми объектами)	Инструкция goto, инструкции break без метки, инструкции break с меткой (частично), поддержка try/catch, блок else (исключение), легковесные процессы (Coroutines)	First class functions, анонимные функции, лексические замыкания, каррирование	Open-source компилятор (интерпретатор), возможность компиляции, интерпретатор командной строки, условия компиляции
PHP	Императивная, объектно-ориентированная, рефлексивная, обобщенное программирование, декларативная, функциональная (частично)	Динамическая, неявная, неявное приведение типов без потери данных, неявное приведение типов с потерей данных, неявное приведение типов в неоднозначных ситуациях, параметрический полиморфизм, информация о типах в runtime, информация о типах-параметрах в runtime	Сборка мусора (освобождения памяти в куче, занятой неиспользуемыми объектами)	Инструкция goto (частично), инструкции break без метки, инструкции break с меткой (частично), поддержка try/ catch, блока finally	Анонимные функции, лексические замыкания	Open-source компилятор (интерпретатор), возможность компиляции, интерпретатор командной строки, условия компиляции

Наименование языка	Парадигмы	Типизация	Управление памятью	Управление потоком вычислений	Функциональные возможности	Компилятор / интерпретатор
Python	Императивная, объектно-ориентированная, функциональная, рефлексивная, обобщенное программирование декларативная, распределенная (ограничено)	Динамическая, неявная, неявное приведение типов без потери данных, информация о типах в runtime, информация о типах-параметрах в runtime	Сборка мусора (освобождения памяти в куче, занятой неиспользуемыми объектами)	Инструкции break без метки, поддержка try/catch, блок finally, блок else (исключение), ленивые вычисления, легковесные процессы (Coroutines)	First class functions, анонимные функции (частично), лексические замыкания, частичное применение, каррирование	Open-source компилятор (интерпретатор), возможность компиляции, Bootstrapping, интерпретатор командной строки
Ruby	Императивная, объектно-ориентированная, функциональная, рефлексивная, обобщенное программирование декларативная, распределенная (ограничено)	Динамическая, неявная, неявное приведение типов без потери данных, информация о типах в runtime, информация о типах-параметрах в runtime	Сборка мусора (освобождения памяти в куче, занятой неиспользуемыми объектами)	Инструкция goto (ограниченно), Инструкции break без метки, инструкции break с меткой, поддержка try/catch, блок finally, блок else (исключения), перезапуски, ленивые вычисления (ограничено), легковесные процессы (Coroutines)	First class functions, анонимные функции, лексические замыкания, частичное применение, каррирование	Open-source компилятор (интерпретатор), возможность компиляции, Bootstrapping, интерпретатор командной строки

функционал достаточно несовершенен, хотя и имеет весьма неплохой синтаксис, отчего код хорошо читается [12].

### Заключение

В заключение отметим, что для успешного и удобного проектирования и написания экспертной системы посредством комплекта языков веб-программирования оптимально все же использовать язык JavaScript как на клиентской, так и на серверной стороне. Указанный язык может быть лишь ограничен браузерными назначениями, при этом он подходит для решения сложных задач в разных ситуациях при наличии таких качеств, как простота и абстракция. Браузер просмотра web-страниц в данном случае выполняет поручения клиентских приложений в вычислительной сети на компьютере пользователя, когда на сервере запускаются серверные приложения.

### Литература

1. Хэррон Д. Node.JS. Разработка серверных веб-приложений в JavaScript / пер. с англ. А.А. Слинкина. М.: ДМК Пресс, 2017. 144 с.
2. Джарратано Д., Райли Г. Экспертные системы. Принципы разработки и программирования / пер. с англ. К.А. Птицына. М.: Вильямс, 2017. 1152 с.
3. Тормозов В.С. Многокритериальная настройка усовершенствованного алгоритма детектирования объектов на изображении // Передовые инновационные разработки. Перспективы и опыт использования, при внедрении в производство: сб. науч. ст. по итогам Восьмой междунар. науч. конф. 2019. С. 35–36.
4. Панкратов С.А. Использование графической информации для защиты программного и информационного обеспечения // Инженерный вестник Дона. 2012. № 2. URL: <http://ivdon.ru/magazine/archive/n2y2012/792> (дата обращения: 25.01.2020).
5. Уилсон Э. Мониторинг и анализ сетей. Методы выявления неисправностей. М.: Лори, 2016. 480 с.
6. Аладышев О.С., Овсянников А.П., Шабанов Б.М. Развитие корпоративной сети Межведомственного суперкомпьютерного центра. URL: <http://vbakanov.ru/metods/1441/corpor.pdf> (дата обращения: 25.01.2020).
7. Гайдук А.Р. Теория и методы аналитического синтеза систем автоматического управления



- (полиномиальный подход). М.: Физматлит, 2017. 264 с.
8. Золкин А.Л. Основы теории систем автоматического управления и автоматизация ЭПС: конспект лекций. Саарабрюкен: LAP LAMBERT Academic Publishing, 2014. 219 с.
  9. Смирнов А.В. Руководство по захвату сетевого трафика. URL: <https://blog.packet-foo.com/2016/11/the-network-capture-playbook-part-3-network-cards> (дата обращения: 25.01.2020).
  10. Perlman R. Interconnections: Bridges & Routers. New York: Addison-Wesley, 2016. 245 p.
  11. Khalil H.K. Nonlinear Systems. 3rd ed. New Jersey: Prentice Hall, 2016. 766 p.
  12. Melin P., Castillo O. Modeling, Simulation and Control of Non-linear Dynamical Systems: An Intelligent Approach Using Soft Computing and Fractal Theory. Boca Raton: Taylor & Francis, 2017. 26 p.

*Получено 14.02.2020*

**Василенко Константин Александрович**, преподаватель Колледжа сервиса и дизайна Владивостокского государственного университета экономики и сервиса. 690092, Российская Федерация, Приморский край, г. Владивосток, ул. Добровольского, 20. Тел. +7 964 453-06-36. E-mail: [k2857@mail.ru](mailto:k2857@mail.ru)

**Золкин Александр Леонидович**, к.т.н., доцент кафедры информатики и вычислительной техники Поволжского государственного университета телекоммуникаций и информатики. 443010, Российская Федерация, г. Самара, ул. Л. Толстого, 23. Тел. +7 960 825-68-49. E-mail: [alzolkin@list.ru](mailto:alzolkin@list.ru)

**Абрамов Николай Викторович**, студент Дальневосточного федерального университета (ДВФУ). 690920, Российская Федерация, Приморский край, г. Владивосток, ул. Суханова, 8. Тел. +7 914 373-91-16. E-mail: [nikolay.abramov1990@mail.ru](mailto:nikolay.abramov1990@mail.ru)

**Курганов Даниил Олегович**, студент ДВФУ. 690920, Российская Федерация, Приморский край, г. Владивосток, ул. Суханова, 8. Тел. +7 999 059-37-74. E-mail: [kurganov\\_vl@mail.ru](mailto:kurganov_vl@mail.ru)

### EXPERT SYSTEMS IN COMPUTER NETWORK: CREATION ISSUES DURING WEB DEVELOPMENT

*Vasilenko K.A.<sup>1</sup>, Zolkin A.L.<sup>2</sup>, Abramov N.V.<sup>3</sup>, Kurganov D.O.<sup>3</sup>*

<sup>1</sup> *Vladivostok State University of Economics and Service, Vladivostok, Russian Federation*

<sup>2</sup> *Povolzhskiy State University of Telecommunications and Informatics, Samara, Russian Federation*

<sup>3</sup> *Far Eastern Federal University, Vladivostok, Russian Federation*

*E-mail: k2857@mail.ru, alzolkin@list.ru, nikolay.abramov1990@mail.ru, kurganov\_vl@mail.ru*

This article looks at the issues of expert systems development in data-processing networks during web programming. The correlation of the processing, storage and transmission of information and web applications to the specificities and problematics of expert systems creation is observed. Under these circumstances, there are difficulties associated with network traffic, namely, with its workload, since there is a need for constant transfer of a large amount of information, which subsequently impedes the processes of contacting of clients with servers. In this case, the information can be subjected to threats of the integrity loss of the transmitted information, since there will be lack of required constant synchronization of data between client services and server services. In order to prevent and eliminate this problem the authors suggest to carefully choose a specific set of web-tools and applications. To achieve this, it is necessary to make a selection of and analyze programming languages and determine the language that corresponds to special requirements for creation of expert systems. Herewith, the new features of web applications for the client with daily developing information technologies should be taken into account.

**Keywords:** *computer networks, web-server, programming language, expert systems, web-developments, information, design*

**DOI:** 10.18469/ikt.2020.18.4.09

**Vasilenko Konstantin Alexandrovich**, Service and Design College of the Vladivostok State University of Economics and Service, 20, Dobrovolskogo Street, Vladivostok, Primorsky Krai, 690092, Russian Federation; Highest category lecturer. Tel. +7 964 453-06-36. E-mail: k2857@mail.ru

**Zolkin Alexander Leonidovich**, Povolzhskiy State University of Telecommunications and Informatics, 23, L. Tolstoy Street, Samara, 443090, Russian Federation; Associate Professor of Informatics and Computer Engineering Department, PhD in Technical Science. Tel. +7 960 825-68-49. E-mail: alzolkin@list.ru

**Abramov Nikolai Viktorovich**, Far Eastern Federal University, 8, Sukhanova Street, Vladivostok, Primorsky Krai, 690092, Russian Federation; Student. Tel. +7 914 373-91-16. E-mail: nikolay.abramov1990@mail.ru

**Kurganov Daniil Olegovich**, Far Eastern Federal University, 8, Sukhanova Street, Vladivostok, Primorsky Krai, 690092, Russian Federation; Student. Tel. +7 999 059-37-74. E-mail: kurganov\_vl@mail.ru

### References

1. Herron D. *Node.JS. Development of Server-Side Web Applications in Javascript*. English trans. by A.A. Slinkin. Moscow: DMK Press, 2017, 144 p. (In Russ.)
2. Dzharratano D., Rajli G. *Expert Systems. Development Principles and Programming*. English trans. K.A. Ptitsyn. Moscow: Vil'jams, 2017, 1152 p. (In Russ.)
3. Tormozov V.S. Multi-criteria adjustment of the advanced algorithm for detecting objects in the image. *Peredovye innovatsionnye razrabotki. Perspektivy i opyt ispol'zovaniya, pri vnedrenii v proizvodstvo: sbornik nauchnyh statej po itogam vos'moj mezhdunarodnoj nauchnoj konferentsii*, 2019, pp. 35–36. (In Russ.)
4. Pankratov S.A. Use of graphic information to protect software and information. *Inzhenernyj vestnik Dona*, 2012, no. 2. URL: <http://ivdon.ru/ru/magazine/archive/n2y2012/792> (accessed: 25.01.2020). (In Russ.)
5. Uilson E. *Monitoring and Analysis of Networks. Troubleshooting Methods*. Moscow: Lori, 2016, 480 p. (In Russ.)
6. Aladyshv O.S., Ovsyannikov A.P., Shabanov B.M. Development of the corporate network of the Interdepartmental Supercomputer Center. URL: <http://vbakanov.ru/metods/1441/corpor.pdf> (accessed: 25.01.2020).
7. Gajduk A.R. *Theory and Methods of Analytical Synthesis of Automatic Control Systems (Polynomial Approach)*. Moscow: Fizmatlit, 2017, 264 p. (In Russ.)
8. Zolkin A.L. *Fundamentals of the Theory of Automatic Control Systems and Automation of EPS: Lecture Notes*. Saarbrücken: LAP LAMBERT Academic Publishing, 2014, 219 p. (In Russ.)
9. Smirnov A.V. A guide to capturing network traffic. URL: <https://blog.packet-foo.com/2016/11/the-network-capture-playbook-part-3-network-cards> (accessed: 25.01.2020).
10. Perlman R. *Interconnections: Bridges & Routers*. New York: Addison-Wesley, 2016, 245 p.
11. Khalil H.K. *Nonlinear Systems*. 3rd ed. New Jersey: Prentice Hall, 2016, 766 p.
12. Melin P., Castillo O. *Modeling, Simulation and Control of Non-linear Dynamical Systems: An Intelligent Approach Using Soft Computing and Fractal Theory*. Boca Raton: Taylor & Francis, 2017, 26 p.

Received 14.02.2020