

Интернет-журнал «Наукovedение» ISSN 2223-5167 <https://naukovedenie.ru/>

Том 9, №6 (2017) <https://naukovedenie.ru/vol9-6.php>

URL статьи: <https://naukovedenie.ru/PDF/179TVN617.pdf>

Статья опубликована 07.02.2018

Ссылка для цитирования этой статьи:

Бальян С.Г. Выдвижение концепции хранения и передачи данных в распределенной системе электронного здравоохранения // Интернет-журнал «НАУКОВЕДЕНИЕ» Том 9, №6 (2017)

<https://naukovedenie.ru/PDF/179TVN617.pdf> (доступ свободный). Загл. с экрана. Яз. рус., англ.

Исследования проведены с использованием вычислительных ресурсов Ресурсного Центра «Вычислительный центр СПбГУ», а также частично поддержаны грантом РФФИ № N 16-07-01111

УДК 004.42

Бальян Сероб Гургенович

ФГБОУ ВО «Санкт-Петербургский государственный университет», Россия, Санкт-Петербург¹

Аспирант

E-mail: serob.balyan@gmail.com

ORCID: <http://orcid.org/0000-0002-0492-7490>

РИНЦ: http://elibrary.ru/author_profile.asp?id=940557

SCOPUS: <http://www.scopus.com/authid/detail.url?authorId=56295235300>

Выдвижение концепции хранения и передачи данных в распределенной системе электронного здравоохранения

Аннотация. Люди с нарушениями речи и/или письма все еще сталкиваются с трудностями в плане общения, несмотря на нынешнее развитое состояние информационных технологий в мире. Статья посвящена облегчению проблем общения у людей с такими расстройствами с помощью использования современных информационных и коммуникационных технологий.

В статье, для использования людьми с такими расстройствами и их врачами, предлагается концепция системы электронного здравоохранения с функциями хранения и передачи разнородных данных. Приводятся ряд требований к системе, для удовлетворения которых рассматриваются различия существующих технологий хранения и передачи данных. На основе полученных результатов и выдвигается концепция, в которой описываются компоненты системы и связи между ними. В статье обсуждены технологии реализации прототипа такой системы: выбираются хранилища, в зависимости от их характеристик и в соответствии с типом хранимых данных, обсуждаются различия языков программирования с учетом разработки системы и выбирается оптимальный. А также, представлен метод оптимизации хранения данных и результаты проведенных тестов.

Ключевые слова: система электронного здравоохранения; хранение разнородных данных; протоколы передачи данных; модели распределения; брокер сообщений

Введение

Несмотря на развитие информационных и коммуникационных технологий (ИКТ), повсеместное использование вычислительных сетей (ВС) и сетей передачи данных как в

¹ 199034, Россия, Санкт-Петербург, Университетская набережная 7-9

повседневной жизни, так и в сфере здравоохранения, существование методов обмена сообщениями на расстоянии, люди с нарушениями речи и письма все еще испытывают трудности в плане общения. В настоящее время разные виды болезней (связанные не только с трудностями общения) могут быть вылечены на некотором уровне, или состояние больных может быть облегчено с помощью использования ИКТ и электронных инфраструктур коммуникации. Такие решения лежат в основе концепции электронного здравоохранения (E-Health).

Существуют разные E-Health системы альтернативной коммуникации для облегчения общения людей с расстройствами речи и/или письма (например). Но, несмотря на большую значимость и эффективность прямого наблюдения и мониторинга поведения лиц с такими расстройствами [4], в существующих разработках отсутствует единая платформа, которая объединила бы в себе возможность преодоления барьера общения и мобильность, с одной стороны, и возможность проведения поведенческого анализа пациентов и диагностирования со стороны врачей – с другой.

При таких возможностях система должна обеспечить возможность обмена разнородными данными между пользователями, их хранение для дальнейшего анализа или ведения журналов, а также, иерархическая структура пользователей должна иметь возможность предоставления доступа участнику конкретной группы к данным, предназначенным именно для этой группы.

В статье предложена концепция хранения и передачи данных для такой системы и на основе предложенной концепции разработан и протестирован программный прототип электронного здравоохранения, который, по сути, является развитием систем, описанные в [5], [6], [7].

Выдвижение концепции

Конкретизируя вышесказанное, получается, что предлагаемая концепция должна вытекать из того, что система на ее основе должна удовлетворять следующим требованиям:

- Хранение персональных, медицинских и общедоступных данных централизованным образом.
- Возможность умеренного контроля поведения больных, как со стороны врачей, так и родителей.
- Обеспечение удаленного обмена специальными сообщениями независимо от времени и местоположения пользователя.
- Обеспечение доступа к данным с учетом использования мобильных клиентов.

Так как первое требование – это хранение разнородных данных и обеспечение доступом к этим данным централизованным образом, а под возможностью контроля поведением (что является вторым требованием) подразумевается ведение журналов сообщений и предоставления доступа к ним привилегированных пользователей, то для удовлетворения данных требований хорошо подходит модель распределения Hub-and-Spoke [8]. Основная идея Hub-and-Spoke проста. Она проистекает из того понятия, что полный граф с n вершинами должен иметь $n/2 * (n - 1)$ ребер (рис. 1).

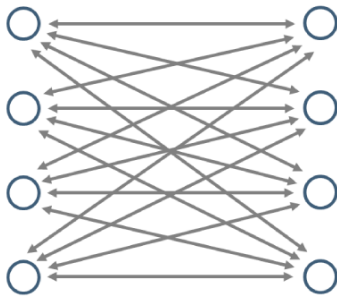


Рисунок 1. Модель распределения Point-to-Point (источник: [9])

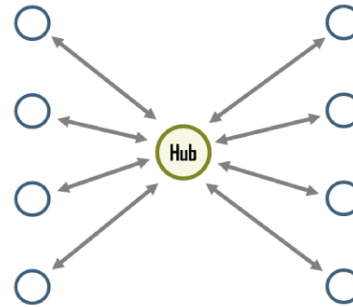


Рисунок 2. Модель распределения Hub-and-Spoke (источник: [9])

Идея состоит в том, что можно избежать квадратного роста числа ребер, если мы сможем заменить «паутину» чем-то наподобие колеса – с центральным узлом и ребрами, выходящими из него ко всем остальным вершинам (рис. 1). В такой топологии число ребер равно числу вершин, что означает, что оно растет линейно в зависимости от числа вершин. При переводе концепции Hub-and-Spoke в реальный мир надо рассмотреть, что из себя на самом деле представляет ребро между двумя вершинами: это может быть очередью сообщений, или темой (topic) модели издатель-подписчик и т. д.

Таким образом, в зависимости от системы, наличие большого количества ребер может стать причиной проблем. Их наличие в большом количестве приведет к большей потребности системы в обслуживании. А небольшое количество маршрутов, наоборот, может привести к более эффективному использованию ресурсов [8].

Но архитектура Hub-and-Spoke также обеспечивает другое существенное преимущество – она разъединяет отправителя и получателя, вставляя активного посредника в центр (hub). Например, центральный узел может выполнять важную функцию маршрутизации входящих сообщений в нужное место назначения. Таким образом, он отделяет отправителя сообщения от необходимости знать местоположение получателя. Передача всех сообщений через центральный компонент также хороша для ведения журнала сообщений или для управления потоком сообщений. Архитектура Hub-and-Spoke, применяемая таким образом, обычно называется брокером сообщений (англ. – Message Broker), потому что центральный узел служит посредником (англ. – broker) для передачи сообщений (англ. – message) между участниками.

В таком подходе центральный узел (сервер) должен связать пользователей между собой и предоставить доступ к соответствующим данным. Получается, что в данном подходе, как и для различных типов телемедицинских систем, фундаментом является архитектура клиент-сервер [10]. Эти данные могут быть личные данные пользователей, их роли, история сообщений, токены доступа, данные о геолокации и т. д. Основная часть данных принадлежит конкретным пользователям, и для осуществления связей между ними должна быть осуществлена иерархическая и реляционная модель хранения. Наилучшей практикой хранения реляционных данных является использование SQL баз данных. Но число обращений к некоторым данным, как токены доступа, учетные данные пользователей, может быть больше чем к другим, поэтому, для ускорения работы сервера и своевременного отклика пользователям, предлагается для часто используемых данных использовать так называемую in-методу базу данных.

Для эффективного использования ресурсов необходимо разделить геолокационные данные от основной базы данных и использовать другое хранилище для них (так как их периодическое отправление от всех пользователей может привести к увеличению времени отклика основной базы данных). Поскольку эти данные сами по себе не имеют иерархической структуры и не содержат никакой вложенной информации, кроме самих геокоординат, нет

необходимости в использовании SQL-хранилищ. Помимо этого, тесты показывают, что некоторые NoSQL документные хранилища быстрее выполняют запросы на запись данных с нерелятивной структурой, чем SQL-хранилища [11]. Следовательно, для задачи хранения геолокационных данных предлагается использовать отдельное NoSQL документное хранилище.

Для хранения файлов было бы более эффективно использовать объектное хранилище, чем файловую систему. Хранение данных в объектно-ориентированной структуре предполагает исключение высокоуровневой абстракции, предоставляемой файловой системой, и ее замену абстракцией объекта. Этот объект, по сути, контейнер, содержащий не только пользовательские данные и стандартные атрибуты, но и метаданные – служебная и дополнительная информация, позволяющая выполнять операции над объектами более эффективно. Поиск файла, сохраненного в виде объекта, можно осуществлять, даже не зная его имени или расширения. Данный подход, с точки зрения удобства, будет более приемлемым, чем, например, параллельная файловая система, описанная в [12], используемая в высокопроизводительных вычислениях.

Так как с такими видами данными должны работать клиенты разных типов, в том числе и мобильные, то нужно создать унифицированный API, который поддерживался бы разными клиентами. API должен быть создан на основе такого метода/протокола передачи данных, который смог бы работать с любым клиентским устройством, и в то же время был легким (с точки зрения количества передаваемых данных) и имел высокую производительность, для лучшей поддержки со стороны мобильных устройств.

Одним из потенциально таких протоколов является SOAP (Simple Object Access Protocol) – простой протокол доступа к объектам. Он позволяет программам, которые работают на разных операционных системах, обмениваться данными с использованием в основном протокола HTTP и расширяемого языка разметки XML [13]. Основными характеристиками SOAP-а являются расширяемость, нейтральность, независимость. Из-за этих характеристик и широкой поддержки HTTP и XML SOAP становится доступным в качестве протокола удаленного доступа практически для любого решения.

В качестве другого варианта можно рассмотреть концепцию взаимодействия между компонентами распределенной вычислительной среды, архитектурный стиль REST (Representational State Transfer) – передача состояния представления [14]. REST-совместимые службы позволяют запрашивающим системам получать доступ к текстовым представлениям ресурсов и манипулировать ими, используя единый и предопределенный набор операций при отсутствии состояния. REST, по сути, является согласованным набором ограничений, учитываемые во время проектирования распределенных систем. Системы, основанные на REST, обеспечивают высокую производительность, надежность и способность расти за счет повторного использования компонентов, которые можно легко изменить или заменить новыми, не затрагивая систему целиком, даже во время ее работы.

В отличие от сервисов, основанных на SOAP, нет никакого официального стандарта для REST API. Это связано с тем, что REST является архитектурным стилем, а SOAP – протоколом. REST не является стандартом сам по себе, но в REST-реализациях используются такие стандарты, как HTTP, JSON и XML.

Несмотря на столь разность, все равно можно сравнивать REST и SOAP по некоторым характеристикам. Тесты, проведенные в компании Oracle, показали, что для мобильных клиентов у REST-а производительность больше, чем у SOAP (рис. 3) [15].

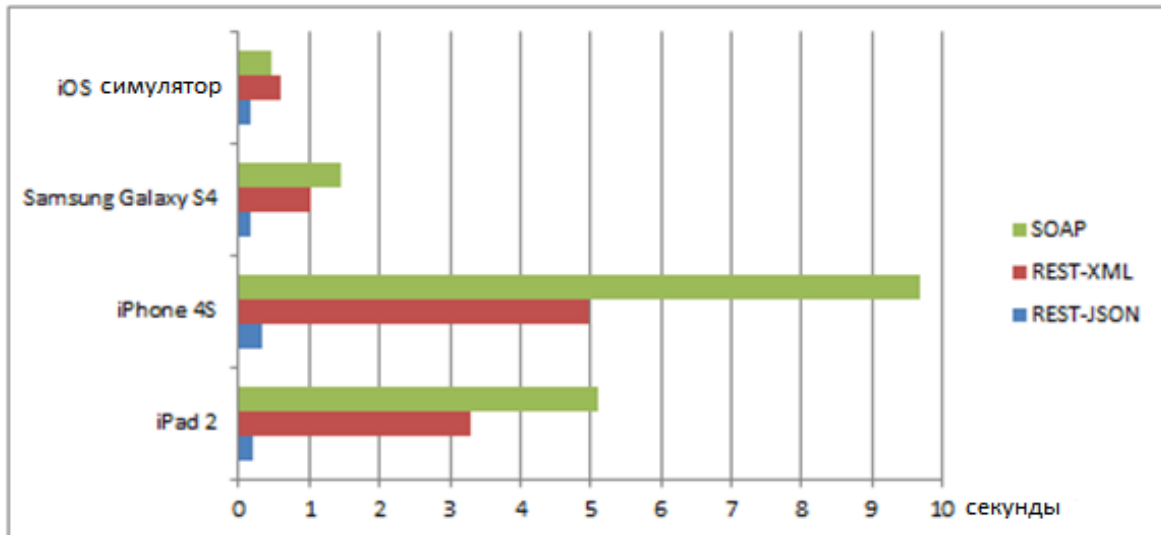


Рисунок 3. Результаты вызовов REST и SOAP служб при применении мобильных клиентов (источник: [15])

Таким образом, для передачи данных между клиентской и серверной частью системы предлагается создать API на основе архитектурного стиля REST.

Суммируя все вышесказанное, предлагается концепция системы электронного здравоохранения с функциями хранения и передачи разнородных данных для людей с расстройствами речи/письма, которую можно описать как совокупность следующих пунктов:

- Использовать SQL базу данных в качестве хранилища опорных данных.
- Для геолокационных данных использовать NoSQL документное хранилище.
- Файловые данные хранить в объектно-ориентированном хранилище.
- Для кэширования часто употребляемых данных использовать базу данных, работающую в оперативной памяти (англ. – in-memory db).
- Для обеспечения связи между пользователями применить модель распределения Hub-and-Spoke.
- С целью обеспечения централизованного доступа к данным, предоставить унифицированный API на основе архитектурного стиля REST (рис. 4).

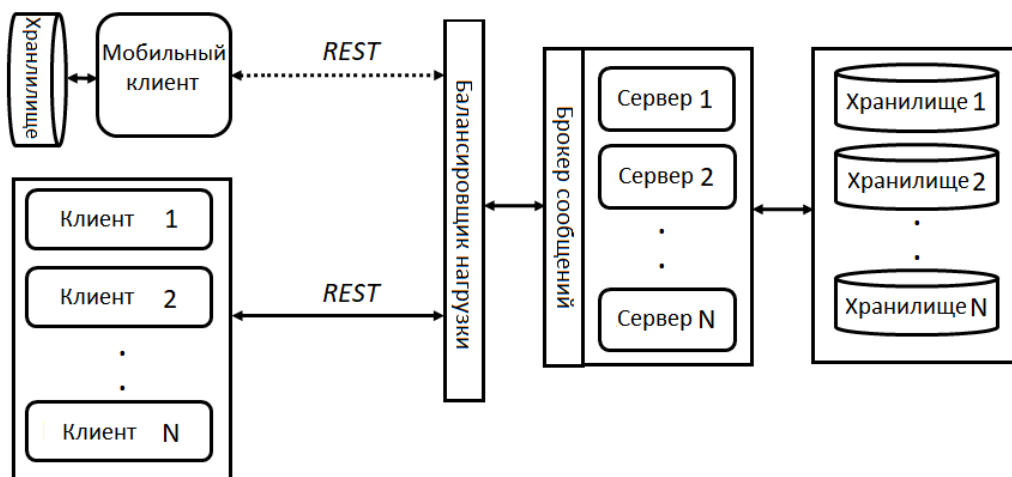


Рисунок 4. Схема выдвигаемой концепции (составлено автором)

Реализация программного прототипа на основе концепции

На основе предложенной концепции был разработан прототип программных модулей и API параллельного взаимодействия, предназначенные для развертывания на большинстве существующих серверных сред (Linux, Windows и т. д.).

Для хранения разнородных данных были рассмотрены ряд существующих хранилищ с открытым исходным кодом, и в качестве опорного SQL хранилища был выбран PostgreSQL², так как данная система легко расширяема и у нее, в отличие от таких соперников как MySQL и MariaDB, более продвинутый механизм обеспечения параллельного доступа [16].

В качестве документных хранилищ были рассмотрены два варианта: MongoDB³ и CouchDB⁴. В MongoDB, в отличие от CouchDB, можно сохранить часть данных в ОЗУ, что увеличивает производительность. Согласно статье [17] MongoDB превосходит CouchDB как по скорости записи, так и по скорости чтения. Поэтому наш выбор был сделан в пользу MongoDB.

В качестве базы данных, работающий в оперативной памяти был выбран Redis⁵. Причина такого выбора в том, что данное хранилище выполняет атомарные обновления путем блокировки и поддерживает асинхронную репликацию. Сообщается, что оно поддерживает около 100 тыс. запросов записи/чтения в секунду на 8-ядерной машине и подходит для обеспечения высокопроизводительных вычислений для небольшого объема данных [18].

OpenStack Swift⁶ был выбран в качестве объектно-ориентированного хранилища для файловых данных.

Тесты, проведенные в компании IBM, показали, что реализация серверного приложения на основе NodeJs имеет более быстрое время отклика, когда уровень параллелизма превышает 50 (рис. 5) [19]. Такой результат можно связать с тем, что характер тестируемого приложения основан больше всего на действиях ввода-вывода, чем на вычислениях, используемых ЦПУ. NodeJs работает с операциями ввода-вывода сравнительно лучше и употребляет меньше ресурсов, чем Java, что подтверждает его назначение: NodeJs был разработан специально для связанных с вводом-выводом приложений и частого обмена данными.

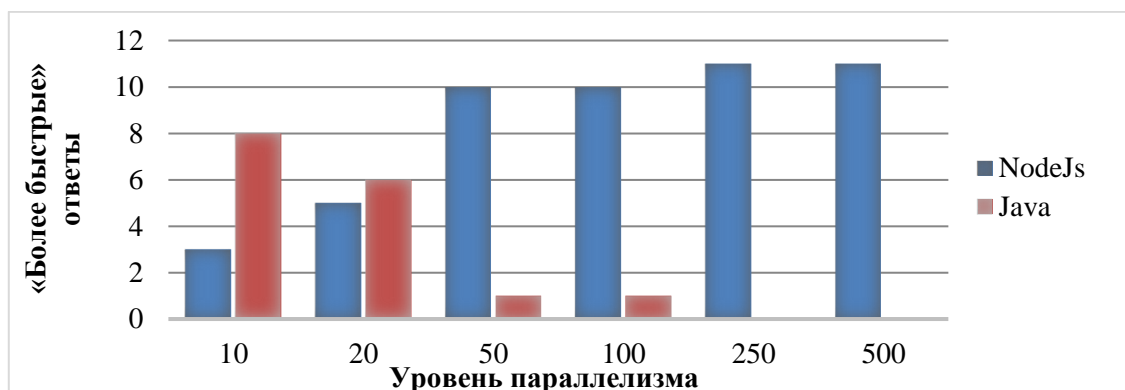


Рисунок 5. Результаты тестирования
(число «более быстрых» ответов) (составлено автором)

² URL: <https://www.postgresql.org/>.

³ URL: <https://www.mongodb.com/>.

⁴ URL: <http://couchdb.apache.org/>.

⁵ URL: <https://redis.io/>.

⁶ URL: <https://wiki.openstack.org/wiki/Swift>.

Тем самым, NodeJs был выбран в качестве основной технологией серверной среды.

В ходе разработки системы была оптимизирована работа между хранилищами PostgreSQL и Redis. Как уже было сказано, одной из причин выбора PostgreSQL в качестве основного реляционного хранилища являлась расширяемость данной системы. В данной системе предоставляется возможность написания собственных расширений, изменяющих поведение системы и/или подстраивающих некоторые функции под внешнее программное окружение. Было написано расширение, которое соединило PostgreSQL с Redis, после чего PostgreSQL смог обращаться к таблицам Redis как к собственным. были проведены одинаковые тесты (вычисление времени отклика в зависимости от количества запросов), как при таком соединении компонентов, так и при использовании серверного приложения в качестве соединяющего звена. Тесты показали, что в данном случае максимальное возможное число запросов в секунду выросло примерно на десять процентов и составило ~2050 запрос/с (рис. 6).

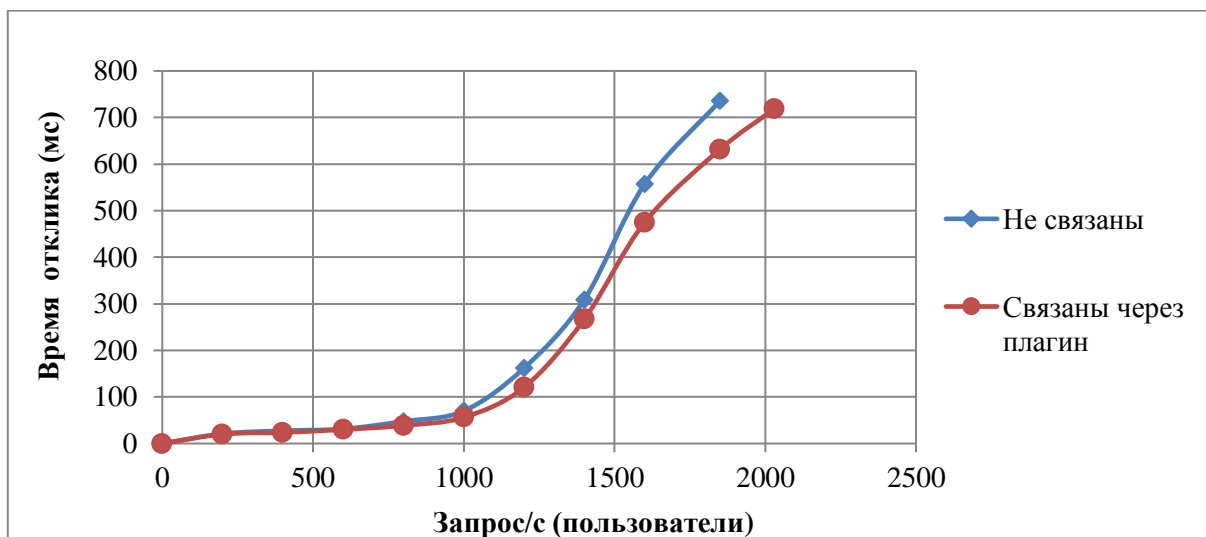


Рисунок 6. Зависимость времени отклика от количества запросов в секунду (составлено автором)

Таким образом можно увеличить количество одновременно используемых систему пользователей.

Заключение и дальнейшая работа

В пределах данной работы были сравнены некоторые существующие модели взаимодействия в распределенных системах, методы хранения и протоколы передачи данных. На основе результатов сравнения предложена концепция архитектуры серверной среды распределенной системы электронного здравоохранения и методов взаимодействия в такой системе для использования людьми с ограниченными возможностями. Право существования такой концепции подтвердилось работоспособностью разработанного на ее основе прототипа, который предназначен для развертывания на большинстве существующих серверных сред. В ходе разработки прототипа был также создан. Проведено экспериментальное исследование эффективности разработанных решений, и на основе полученных результатов осуществлена оптимизация методов хранения данных.

В дальнейшем планируется довести прототип до финальной версии, а также добавить алгоритмы искусственного интеллекта для анализа данных и повышения удобства использования как со стороны больных (например, подсказывание следующего слова при

наборе текста), так и врачей и, естественно, учитывать особенности хранения специальных словарей для таких алгоритмов.

ЛИТЕРАТУРА

1. Mitchell J. From telehealth to e-health: The unstoppable rise of e-health, Commonwealth Department of Communications, Information Technology and the Arts (DOCITA), Canberra, Australia, 1999.
2. Keskinen T., Heimonen T., Rajaniemi J., Kauppinen S. SymbolChat: Picture-Based Communication Platform for Users with Intellectual Disabilities // Lecture Notes in Computer Science, Vol. 7383, 2012. pp. 279-286.
3. Takasaki T., Mori Y. Design and Development of a Pictogram Communication System for Children Around the World // Lecture Notes in Computer Science, Vol. 4568, 2007. pp. 193-206.
4. Eikeseth S., Eikeseth L., Jahr E., Karlsson P. Outcome for children with autism receiving early and intensive behavioral intervention in mainstream preschool and kindergarten settings // Research in Autism Spectrum Disorders, Vol. 6, 2012. pp. 829-835.
5. Абрамян С., Балян С., Мурадов А. Программное обеспечение совместного принятия решений на основе мобильных инфраструктур // Процессы управления и устойчивость, Т. 2, № 18, 2015. С. 333-339.
6. Abrahamyan S., Balyan S., Muradov A., Kulabukhova N., Korkhov V. A Concept of Unified E-Health Platform for Patient Communication and Monitoring // Lecture Notes in Computer Science, Vol. 10408, 2017. pp. 448-462.
7. Abrahamyan S., Balyan S., Muradov A., Korkhov V., Moskvicheva A., Jakushkin O. Development of M-Health software for people with disabilities // Lecture Notes in Computer Science, Vol. 9787, 2016. pp. 468-479.
8. Hohpe G. Hub and Spoke [or] Zen and the Art of Message Broker Maintenance // Enterprise Integration Patterns. 2003. URL: http://www.enterpriseintegrationpatterns.com/ramblings/03_hubandspoke.html (дата обращения: 12.12.2017).
9. Forattini C. Boeing Still Underpriced Despite Recent Rally // Seeking Alpha. 2017. URL: <https://seekingalpha.com/article/4063143-boeing-still-underpriced-despite-recent-rally> (дата обращения: 12.11.2017).
10. Гуськов В., Гушанский Д., Кулабухова Н., Абрамян С., Балян С., Дегтярев А., Богданов А. Интерактивный инструментарий для распределенных телемедицинских систем // Компьютерные исследования и моделирование, Т. 7, № 3, 2015. С. 521-527.
11. Mohammed J. Is Postgres NoSQL Better Than MongoDB? // Aptuz technology solutions. 2015. URL: <http://www.aptuz.com/blog/is-postgres-nosql-database-better-than-mongodb/> (дата обращения: 12.12.2017).
12. Ганкевич И., Балян С., Абрамян С., Корхов В. Применение создаваемых по требованию виртуальных кластеров в высокопроизводительных вычислениях // Компьютерные исследования и моделирование, Т. 7, № 3, 2015. С. 511-516.
13. Вох , Ehnebuske D., Kakivaya G., Layman A., Mendelsohn N. Simple Object Access Protocol (SOAP) 1.1 // Официальный веб-сайт World Wide Web Consortium. 2000.

- URL: <https://www.w3.org/TR/2000/NOTE-SOAP-20000508/> (дата обращения: 12.12.2017).
14. Fielding T. R. Representational State Transfer (REST) // Официальный веб-сайт Donald Bren School of Information and Computer Sciences. 2000. URL: https://www.ics.uci.edu/~fielding/pubs/dissertation/rest_arch_style.htm (дата обращения: 12.12.2017).
 15. Davelaar S. Performance Study – REST vs SOAP for Mobile Applications // Oracle: A-Team Chronicles. 2015. URL: <http://www.ateam-oracle.com/performance-study-rest-vs-soap-for-mobile-applications/> (дата обращения: 12.12.2017).
 16. Conrad T. PostgreSQL vs. MySQL vs. Commercial Databases: It's All About What You Need // Веб-сайт devx.com. 2004. URL: <http://www.devx.com/dbzone/Article/20743> (дата обращения: 12.12.2017).
 17. Li Y., Manoharan S. A performance comparison of sql and nosql databases // IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM). Victoria, B. C., Canada. 2013. pp. 15-19.
 18. Cattell R. Scalable SQL and NoSQL data stores // Special Interest Group on Management of Data of the Association for Computing Machinery (ACM SIGMOD) Record, Vol. 39, No. 4, December 2011. pp. 11-27.
 19. Cross Z., Santiago D., Singh D. Developing mobile apps with Node.js and MongoDB, Part 1. A team's methods and results // Официальный веб-сайт IBM. 2013. URL: <https://www.ibm.com/developerworks/library/mo-nodejs-1/#N10191> (дата обращения: 12.12.2017).

Balyan Serob Gurgenovich

Saint Petersburg university, Russia, Saint Petersburg

E-mail: serob.balyan@gmail.com

Suggestion of a concept of data storing and transferring in a distributed E-Health system

Abstract. People with speech and/or writing disorders still face communication difficulties, despite the advanced state of modern information technologies. The article is devoted to the facilitation of the communication problem of such people with the help of modern information and communication technologies.

The concept of an E-Health system with the functions of storing and transferring heterogeneous data for use by people with such disorders and their doctors is proposed in the article. A number of initial requirements to the system are given, to meet which number of data transfer and storage technologies are examined. Based on the results obtained, a concept is proposed, which describes the system components and the relationships between them. Technologies for developing a prototype of such a system are described: the storages are selected, depending on their characteristics and in accordance with the type of stored data, the differences in the programming languages are discussed and the optimal one is chosen. Also, data storing optimization method is described and the test results are shown.

Keywords: E-Health system; storage of heterogeneous data; data transfer protocols; distribution models; message broker