

$$TE_{ij} = \frac{indicator_Te_{ij}}{\max(indicator_Te_{ij})}, i=1..n; j=1..k, \quad (3)$$

где $indicator_Te_{ij}$ – частный i -ый показатель технологической эффективности j -ой ИС в ЛД;
 $\max(indicator_Te_{ij})$ – максимальный по значению частный i -ый показатель технологической эффективности j -ой ИС в ЛД.

Частные показатели технологической эффективности $indicator_Te_{ij}$ предлагаем определять методом подсчета количества бинарных оценок, отражающих факт наличия или отсутствия («0/1») соответствующего показателя.

$$EEI_j = \sqrt{\frac{1}{m} \sum_{l=1}^m EE_{lj}^2}, \quad (4)$$

где EE_{lj} – частный l -ый индекс экономической эффективности j -ой информационной системы в ЛД;
 m – количество l -ых частных индексов экономической эффективности j -ой информационной системы в ЛД.

$$EE_{lj} = \frac{indicator_Ee_{lj}}{\max(indicator_Ee_{lj})}, \quad (5)$$

$$EE_{lj} = \frac{\min(indicator_Ee_{lj})}{indicator_Ee_{lj}}, \quad (6)$$

где $indicator_Ee_{lj}$ – частный l -ый показатель экономической эффективности j -ой информационной системы в ЛД;
 $\max(indicator_Ee_{lj})$ – максимальный по значению частный l -ый показатель экономической эффективности j -ой информационной системы в ЛД;

$\min(indicator_Ee_{lj})$ – минимальный по значению частный l -ый показатель экономической эффективности j -ой информационной системы в ЛД.

Для оценки частных показателей экономической эффективности $indicator_Ee_{lj}$ рекомендуется применять финансовые методы инвестиционных ИТ-проектов [3]. Для относительного сопоставления финансовых показателей нескольких ИТ-проектов и расчета частных индексов экономической эффективности EE_{lj} в зависимости от экономического смысла показателей предлагаем использовать одну из моделей (5-6).

Новизна и отличительные особенности сформированной системы показателей (рис.1) и системы экономико-математических моделей (1-6) заключаются в интеграции системного, комплексного и индексного подходов, гибкости и простоты.

Заключение. Практическая значимость полученных результатов исследования состоит в возможности их использования при разработке методики для комплексной оценки эффективности современных информационных технологий и систем в ЛД и обосновании управленческого решения по выбору наилучшей логистической информационной системы.

1. Концепция развития логистической системы Республики Беларусь на период до 2030 года [Электрон. ресурс]. – Режим доступа: <http://www.government.by/ru/solutions/3083>. – Дата доступа: 25.02.2019.
2. Республиканская программа развития логистической системы и транзитного потенциала на 2016-2020 годы [Электрон. ресурс]. – Режим доступа: <http://government.by/upload/docs/file10f0af8923c585e3.PDF>. – Дата доступа: 25.02.2019.
3. Ткалич, Т.А. Экономическая эффективность информационных систем. Методология оценки ключевых показателей и ее практические приложения / Т.А. Ткалич. – Saarbrücken : LAP LAMBERT Academic Publishing, 2013. – 329с.
4. Логистическая информационная система [Электрон. ресурс] // TADVISER, 2005-2019. – Режим доступа: <http://www.tadviser.ru/>. – Дата доступа: 25.02.2019.
5. Забродская, К.А. Модели и методическое обеспечение оценки уровня развития инфокоммуникационных услуг в Республике Беларусь: дис. ... к-та экон. наук: 08.00.13 / К.А. Забродская. – Минск, 2015. – 180 с.

ОСОБЕННОСТИ РАЗРАБОТКИ WEB-ПРИЛОЖЕНИЙ РЕАЛЬНОГО ВРЕМЕНИ НА ОСНОВЕ .NET И SIGNALR

Кленча А.И.,

студент 4 курса ВГУ имени П.М. Машерова, г. Витебск, Республика Беларусь

Научный руководитель – Семенов М.Г., канд. физ.-мат. наук

В большинстве web-приложений или ресурсов (таких как Twitter, Instagram и прочее) присутствуют элементы интерфейса, содержимое которых во время работы должно постоянно иметь актуальную информацию. В роли таких элементов могут выступать уведомления о событиях (например, уведомления о сообщениях в чате), счётчики, индикаторы и прочее. Приложения, содержащие такие элементы, называют приложениями реального времени. В процессе разработки web-приложений реального времени возникает следующий вопрос: показывать актуальную информацию после обновления страницы либо реализовать автоматическое обновление данных в необходимых элементах? Наиболее перспективным на текущий момент является реализация динамического обновления контента.

При разработке приложений для небольшой аудитории пользователей, где система не будет подвержена большим нагрузкам, приемлемым вариантом является использование технологии AJAX. Логика данного

решения заключается в постоянном опросе клиентом сервера на наличие обновленной информации, и, если такие изменения существуют, JavaScript клиентской стороны отобразит полученные данные. Однако для проектов, у которых количество одновременных подключений пользователей превышает несколько тысяч, будет создаваться слишком огромная нагрузка на серверное оборудование. Одним из способов решения данной проблемы является использование прикладных библиотек специального назначения.

SignalR – это прикладная библиотека от Microsoft .NET для обеспечения асинхронного клиент-серверного взаимодействия в режиме реального времени с помощью содержимого push-уведомлений.

Целью данной работы является изучение библиотеки SignalR и принципов построения с ее помощью приложений реального времени, а также создание тестового web-приложения, демонстрирующего возможности библиотеки.

Материал и методы. Для изучения библиотеки используется среда разработки Microsoft Visual Studio 2017, фреймворк ASP .NET MVC 5, а также сама библиотека SignalR, подключаемая к проекту через систему управления пакетами Nuget. Методы исследования – методы объектно-ориентированного программирования, системный анализ, шаблоны проектирования.

Результаты и их обсуждение. Библиотека SignalR представляет собой API для обеспечения связи клиент-сервер посредством удаленного вызова процедур (RPC). RPC вызывают функции на языке JS клиентской стороны из кода приложений .NET серверной части [1].

К основным функциям SignalR можно отнести:

- автоматическое определение, обработка и управление подключениями;
- поддержка широковещания на всех подключенных клиентов одновременно;
- создание групп пользователей, что позволяет сделать рассылку уведомлений по отдельным категориям более удобной;
- масштабирование системы.

Для обеспечения обмена данными библиотека SignalR автоматически определяет наиболее подходящий «транспорт», однако при необходимости разработчики могут явно указывать желаемый способ обеспечения взаимодействия. К поддерживаемым типам технологий относятся [2]:

- WebSockets.
- Server Sent Events.
- Forever Frames.
- Long Polling.

Схема обеспечения клиент-серверного взаимодействия при использовании библиотеки SignalR представлена на рисунке 1.

Основой коммуникаций в SignalR являются концентраторы (Hubs). Они обслуживают входящие запросы к серверу и обеспечивают рассылку сообщений клиентам либо с самого концентратора, либо от лица другого пользователя системы. Он представляет собой шлюз для обмена данными. Кроме того, помимо обеспечения связи между клиентами и передачи их сообщений, концентратор позволяет реализовать рассылку системных уведомлений, что крайне удобно для реализации систем контроля [3] (например, системы голосования или оповещения об изменениях в системе).

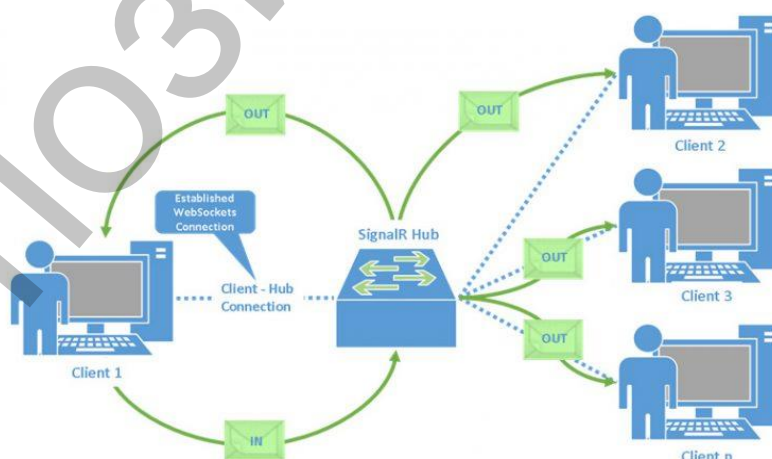


Рисунок 1 – Схема клиент-серверного взаимодействия в приложении с использованием библиотеки SignalR

Работа с библиотекой SignalR на клиентской стороне обеспечивается с помощью вызова функций JavaScript. Для взаимодействия с концентратором, достаточно получить прокси концентратора и опреде-

лить ряд функций, которые будут обеспечивать функционал приложения, либо вызывать те методы, которые были определены в самом концентраторе.

Заключение. При выполнении работы была рассмотрена библиотека SignalR и создано тестовое web-приложение, демонстрирующее обмен данными в режиме реального времени на примере чата. В дальнейшем реализованное приложение будет использоваться при проектировании информационного портала для обеспечения информационной связи между пользователями системы.

1. Введение в SignalR 2 // Metanit [Электронный ресурс]. – Режим доступа: <https://metanit.com/sharp/mvc5/16.1.php>. – Дата доступа: 20.02.2019.
2. Microsoft Docs [Электронный ресурс]. – Режим доступа: <https://docs.microsoft.com/ru-ru/aspnet/signalr/overview/getting-started/introduction-to-signalr>. – Дата доступа: 21.02.2019.
3. Appel, R. Modern Apps - Use SignalR to Build Modern Apps / R. Appel // MSDN Magazine – 2014. – № 10. – P. 52–55.

РЕАЛИЗАЦИЯ ГЕНЕТИЧЕСКИХ АЛГОРИТМОВ НА ОСНОВЕ КОМПЬЮТЕРНЫХ МОДЕЛЕЙ

Кравцов К.В.,

учащийся 4 курса Оршанского колледжа ВГУ имени П.М. Машерова,

г. Орша, Республика Беларусь

Научный руководитель – Трибис А.В.

Генетические алгоритмы – это достаточно известные в данный момент времени методы решения задач оптимизации. Они построены на использовании эволюционных принципов для поиска оптимального решения. Данная идея вызывает довольно широкий интерес, чтобы её реализовать.

Генетические алгоритмы чаще всего используются для решения задач оптимизации, т.е. задач, в которых есть какая-то функция двух, трех, четырех или более переменных: $F(x_1, x_2, \dots, x_n)$ и требуется найти максимум данной функции или же минимум. Функция F называется целевой функцией, а её переменные – параметрами. Генетические алгоритмы «прикреплены» к этой задаче. А именно, переменные функции – это генетический материал или же гены. Соединение одинаковых генов – это хромосома. Каждая особь имеет свою хромосому, а, значит, у нее есть свой набор параметров. Взяв целевую функцию и вставив туда эти параметры, получится n -ое значение. Далее оценивается полученное значение, соответствует ли оно условиям, которые были поставлены в начале – это и будет определять характеристику особи. Ключевой этап в работе генетического алгоритма – скрещивание особей между собой естественным путем при помощи операторов, в результате скрещивания мы имеем потомство и часть этих потомков замещает старое поколение.

Цель исследования – на основе компьютерной модели создать программу, позволяющую демонстрировать работу генетических алгоритмов.

Материал и методы. Для демонстрации модели реализации генетических алгоритмов была разработана программа «ReLIFE» в среде разработки Visual Studio, язык программирования C#.

Результаты и их обсуждение. Во время разработки ПО было составлено техническое задание, непосредственно заказчиком данного продукта и на основании этого был составлен список этапов разработки программы: изучение предметной области, а также определение с языком программирования; составление диаграмм, таких как: диаграмма вариантов использования, диаграмма последовательности, диаграмма коопераций, диаграмма классов, диаграмма компонентов, диаграмма развертывания, диаграмма состояний, диаграмма деятельности; создание кода; редактирование кода; тестирование.

В процессе создания приложения было протестировано 10 модулей: «Открытие главного окна, первый раз», «Переход по вкладкам приложения», «Функция кнопки запуска симуляции», «Функция кнопки записи Gif анимации», «Функция кнопки остановки симуляции», «Функция добавления записи в библиотеку», «Функция редактирования записи в библиотеке», «Функция удаления записи из библиотеки», «Загрузка библиотеки при повторном запуске», «Функция кнопки «Help», «Функция кнопки «Exit». Все 11 модулей успешно прошли проверку и программное средство готово к внедрению.

В приложении реализована симуляция жизни простейших организмов на основе генетического алгоритма, возможность создавать gif-анимации, создавать статьи, которые автоматически помещаются в библиотеку, сохранять и загружать данные библиотеки.

На рисунке представлена экранная форма программы «ReLIFE» в разных режимах работы.

В результате исследования, было разработано приложение, демонстрирующее реализацию эволюционного моделирования посредством генетических алгоритмов. Программа включает в себя 3 модуля: симуляция генетических алгоритмов, библиотека, позволяющая записывать симуляцию и добавлять комментарии, подписи, а также справка по использованию программного средства.