

УДК 681.3

В. А. Игнатюк¹, В. М. Нагаев², С. И. Сметанин³

РАЗРАБОТКА ВЕБ-ИНТЕРФЕЙСА СИСТЕМЫ МОНИТОРИНГА ТРАНСПОРТА НА БАЗЕ GPS/ГЛОНАСС

В работе осуществлена разработка составной части системы мониторинга, работающей в двух вариантах: в виде ГИС-клиента и веб-приложения. В совокупности друг с другом они образуют более гибкое программное обеспечение, способное работать в той или иной сфере применения в зависимости от предпочтений конкретного клиента. Оптимизирована прокладка маршрута. Проработана система навигационных подсказок.

Ключевые слова: *трек, сетевой протокол, веб-интерфейс, GPS/ГЛОНАСС, цифровая сеть, широкополосная связь, точка доступа, сервер, типы серверов, арифметические команды, память программ, спутниковая связь, спутниковый мониторинг.*

Введение

Возможности, предоставляемые системами глобального позиционирования GPS/ГЛОНАСС, позволяют удовлетворить интересы широкого круга потребителей. На настоящий момент многие компании прибегают к услугам спутникового мониторинга для контроля и обеспечения безопасности [1].

Между тем, у существующего программного обеспечения систем мониторинга имеется множество недостатков.

Во-первых, карты, используемые в таких системах, лишь частично информируют пользователя о местонахождении объекта. Опускается часть важных объектов, например, таких, как магазины, школы, общественные центры. Это приводит к недостаточной информированности о положении объекта мониторинга и, как следствие, является основным недостатком используемых карт.

¹ © Виктор Александрович Игнатюк, д-р физ.-мат. наук, профессор кафедры электроники Владивостокского государственного университета экономики и сервиса, ул. Гоголя, 41, г. Владивосток, Приморский край, 690014, Россия, E-mail: viktor.ignatyuk@vvsu.ru.

² © Владимир Михайлович Нагаев, аспирант кафедры электроники Владивостокского государственного университета экономики и сервиса, ул. Гоголя, 41, г. Владивосток, Приморский край, 690014, Россия.

³ © Станислав Игоревич Сметанин, студент кафедры электроники Владивостокского государственного университета экономики и сервиса, ул. Гоголя, 41, г. Владивосток, Приморский край, 690014, Россия.

Во-вторых, остаются нерешенными некоторые проблемы навигации. Используемые алгоритмы прокладки маршрута часто не учитывают наличие грузового транспорта, для которого часть дорог является не рекомендуемой к использованию, не учитывают наличие пешеходов, которые могут использовать не только дороги, открытые для автомобилей, но и узкие аллеи, тропы и пр., что существенно повышает число возможных маршрутов [2].

Наконец, интерфейс, доступный пользователю, зачастую не позволяет получать информацию о различных объектах на карте, в том числе отсутствуют «подсказки», что существенно позволило бы упростить пользовательскую навигацию на карте.

На решение данных проблем направлен создаваемый веб-интерфейс.

Описание системы

Система состоит из 3 основных компонентов: TCP-сервера, разрабатываемого на языке С под операционную систему Linux, базы данных под управлением СУБД PostgreSQL 9.0 и WebForm, созданных по технологии ASP.NET (рис. 1). В качестве веб-сервера может использоваться как Internet Information Server (IIS), поддерживаемый компанией Майкрософт, так и любой другой сервер, способный интерпретировать код ASP.NET, например, такой популярный сервер, как Apache.

Основным преимуществом веб-технологий считается их доступность [4]. Данная разработка позволяет осуществлять мониторинг за объектом с любого компьютера, через любой веб-браузер, что делает удобным использование системы, поскольку не возникает необходимости в установке дополнительного программного обеспечения.

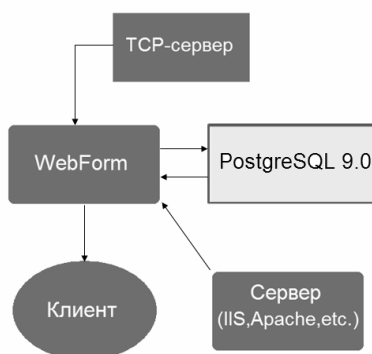


Рис. 1. Описание системы

Таким образом, разрабатываемый веб-интерфейс позволяет отслеживать перемещение объекта на 2D-карте с различными уровнями масштабирования, определяемыми пользователем, прокладывать оптимальный

маршрут к заданной точке прибытия, проставлять контрольные точки маршрута, осуществлять поиск по карте, строить графики расхода топлива и скорости объекта в заданные промежутки времени.

Карты

Система работает с картами в польском формате. Польский формат – это текстовый формат хранения данных, разработанный компанией GPSPMapper, позволяющий легко дополнять и редактировать себя, что делает его удобным для создания картографического программного обеспечения со специфическими требованиями.

Структура польского формата позволяет хранить картографические данные с разной степенью детализации в зависимости от масштаба отображения карты. Для этого используется понятие уровня [3]. Каждый уровень имеет свой диапазон масштаба, который не пересекается с другими уровнями. В карте, используемой при создании проекта, насчитывается всего 5 уровней. Самый первый уровень, нулевой, соответствует максимальному масштабированию карты, при котором можно различать отдельные дома. Уже на первом уровне видны лишь отдельные улицы и так далее до 4-го уровня, на котором отображается, например, весь Приморский край.

Польский формат позволяет создавать карты практически любой сложности. Основными объектами карты являются полигоны, линии и точки.

Полигоны – это замкнутые площадные объекты, используемые для отображения таких типов объектов, как моря, озёра, леса, болота и пр.

Линии – это незамкнутые линии произвольной геометрии, ими отображаются дороги, ручьи, тропы, просеки и пр.

Точки – соответственно точечные объекты для отображения маяков, заправок, бродов, глубин и пр.

Используемые в данном проекте карты прекрасно документированы и содержат огромное количество различных объектов: продовольственные магазины, автомастерские, школы, кафе и др., что позволяет с высокой эффективностью осуществлять мониторинг за выбранными объектами.

WebForm и ASP.NET

Для разработки веб-интерфейса была выбрана технология ASP.NET.

В основе ASP.NET – платформа Microsoft .NET. Эта платформа представляет собой среду выполнения программ, по сути – виртуальную машину для языка MSIL, причем перед исполнением код MSIL преобразуется в машинный код, что обеспечивает высокую скорость исполнения.

Использование .NET позволяет нам применять для разработки любые языки этой платформы – в данной системе используется C#. Этот язык применяет механизм автоматического управления памятью, так называемую сборку мусора (garbage collection), что предотвращает от таких (ти-

пичных для языков с ручным освобождением памяти) ошибок программирования, как утечки памяти (memory leaks).

Кроме того, выбор платформы .NET позволяет реализовать многоплатформенность клиентов и добиться отображения обстановки в реальном режиме времени.

В работе системы было использовано разделение на клиентскую часть, представленную с помощью JavaScript, и серверную часть, реализованную на языке C#. Работа серверной части заключается в том, что на основании параметров, полученных от клиентской части, сервер формирует изображение в формате .jpg, являющееся требуемой областью карты. Клиентская часть передает серверу различные параметры, например: количество пикселей, на число которых осуществляется перемещение карты по вертикальной или горизонтальной оси, текущие координаты мыши, информацию о масштабировании карты, запросы о выводе подсказки и получении информации о дальности и длительности построенного трека и т.п.

Изображение, формируемое сервером в формате .jpg, разбивается на изображения меньших размеров, образуя «мозаичное изображение». Это позволяет более быстро отображать загружаемую информацию. Подобная технология применяется во всех популярных сервисах карт, например Google Maps.

Если пользователь перемещает или масштабирует карту, срабатывают события JavaScript, меняющие содержимое изображений. Это происходит с помощью динамически формируемого GET-запроса, передающего соответствующие координаты ASP.NET странице сервера, формирующей требуемые изображения для клиента (рис. 2).





Рис. 2. Вид карты при различных уровнях масштабирования

Графика

Для отображения графики используются встроенная библиотека платформы .NET, System.Drawing, поддерживающая работу с GDI+.

Для ускорения работы карты применяется хранение базовых типов объектов в структурах, а не классах, что делает доступ к ним более быстрым за счет устранения затратных процедур упаковки и распаковки.

Кроме того, применяется проверка на попадание геом. фигуры в область карты, что ускоряет работу программы, делая ненужными внутреннюю проверку и обрезание лишней части карты.

Функции навигации

Система поддерживает такие функции навигации, как определение объекта на карте (динамическая подсказка), поиск по заданным параметрам, прокладку оптимального маршрута между произвольно выбранными точками маршрута.

Динамическая подсказка

Для определения объекта на карте используется динамическая подсказка, созданная по технологии AJAX.

AJAX (от англ. Asynchronous Javascript and XML – «асинхронный JavaScript и XML») – подход к построению интерактивных пользовательских интерфейсов веб-приложений, заключающийся в «фоновом» обмене данными браузера с веб-сервером. В результате, при обновлении данных веб-страница не перезагружается полностью, и веб-приложения становятся более быстрыми и удобными (рис. 3) [5].

Это реализовано следующим образом: каждый раз, когда пользователь держит курсор мыши на одном месте более 2-х секунд, на сервер выполняется асинхронный запрос, передающий серверу в качестве параметров координаты мыши пользователя.

На сервере вначале определяется, попадает ли объект в область видимости карты. Если хотя бы одна точка фигуры попадает в границы карты, то считается, что объект находится в пределах видимости карты. Выбранные объекты заносятся в коллекцию типа ArrayList, и затем, при второй проверке, производится более подробная выборка из объектов, состоящая в проверке, лежит ли точка на линии, является ли объект полилинией или находится в пределах фигуры, является ли объект полигоном или точкой.



Рис. 3. Интерфейс веб-формы

Затем по выбранным результатам выясняется, на какой именно объект кликнул пользователь. Проверка основывается на иерархичности объектов по отношению друг к другу. Как самый маленький объект на карте, если координаты мыши находятся в пределах радиуса точки, выбранный объект считается точкой. Далее, если выбранный объект не является точкой, проверяется, лежит ли точка на каком-либо объекте из перечня полилиний, и, если координаты мыши не соответствуют ни одной из полилиний, выбранный объект считается полигоном. Если присутствует множество вложен-

ных друг в друга полигонов, массив с полигонами сортируется в обратном порядке. Это делается на основании того, что на первые прорисованные полигоны накладываются полигоны, прорисованные позднее, таким образом, полигон, прорисованный последним, удовлетворяющий координатам мыши пользователя, будет выбранным полигоном, или самым первым элементом в обратно сортированном массиве полигонов.

Прокладка маршрута

Система также позволяет прокладывать маршрут между контрольными точками на карте по алгоритму Дейкстры с учетом направления дороги, позволяя выбирать между тремя различными режимами: кратчайший маршрут, самый быстрый маршрут и маршрут с учетом класса дороги.

Разница между этими режимами в том, что кратчайший маршрут прокладывается только с учетом расстояния между точками, тогда как самый быстрый маршрут прокладывается исходя из максимально возможной скорости для данной конкретной дороги, т.е. вес ребра графа будет определяться как расстояние/скорость. Это означает, что маршрут прокладывается исходя из минимально затраченного времени на прохождение дороги. Таким образом, дорога, у которой скорость будет выше, будет считаться алгоритмом более приоритетной.

Маршрут с учетом класса дороги прокладывается таким же образом, но учитывается параметр «комфортность» дороги. Предпочтение отдается дороге с более высоким классом дороги: например, между обычной дорогой и автомагистралью, алгоритм остановится на автомагистрали.

Возможен выбор между следующими объектами мониторинга: обычным, грузовым транспортом и пешеходом. Для грузового транспорта некоторые дороги будут запрещены к проезду, в результате чего маршрут будет строиться в обход этих дорог.

При загрузке карты программа автоматически распределяет точки и смежные им точки в отдельные массивы данных, вычисляя расстояние между ними. Затем, при изменении масштаба, координаты просто пересчитываются, умножаясь на некоторый коэффициент масштабирования, соответствующий данному уровню карты.

Во время начала работы алгоритма происходит определение точки с минимальной вершиной: первой выбранной вершиной будет вершина, которая передается как начальная точка алгоритма, остальные вершины принимают значения бесконечности. После этого по сумме ребра и метке точки выбирается наименьшая точка из всех смежных выбранной точке. После того как все смежные точки будут просмотрены, цикл запускается сначала, выбирая начальной точкой ту, которая имеет наименьшую вершину из всех вершин, смежных с предыдущей точкой. Так продолжается либо до того момента, когда все точки будут пройдены – в этом случае

считается, что путь не найден, либо до того момента, когда точка с минимальной вершиной будет равняться конечной точке алгоритма.

Заключение

Данная разработка является составной частью системы мониторинга, работающей в двух вариантах: в виде ГИС-клиента и веб-приложения. В совокупности друг с другом они образуют более гибкое программное обеспечение, способное работать в той или иной сфере применения в зависимости от предпочтений конкретного клиента.

-
1. Соловьев Ю. А. Спутниковая навигация и ее приложения. – М.: Изд-во Эко-Трендз, 2003.
 2. Карлащук В. И., Карлащук С. В. Спутниковая навигация. Методы и средства. – М.: Изд-во «Солон-Пресс», 2006.
 3. GPSPmapper manual: официальная документация по-польскому формату.
 4. Создание приложений Microsoft ASP.NET». – М.: Изд-во «Русская Редакция», 2002.
 5. Matthew MacDonald, Mario Szpuszta Pro ASP.NET 2.0 in C# 2005». – М.: Изд-во «Вильямс», 2008.