

УДК 004.942

МЕТОДЫ И АЛГОРИТМЫ ПОИСКА ОПТИМАЛЬНОГО ПУТИ ПЕРЕМЕЩЕНИЯ АГЕНТОВ ПО ТЕРРИТОРИАЛЬНО-РАСПРЕДЕЛЕННЫМ ТОРГОВЫМ ТОЧКАМ

*Поправко О.О., Телятников А.О.
Донецкий национальный технический университет
кафедра автоматизированных систем управления
E-mail: popravkodon@gmail.com*

В статье рассматривается вопрос нахождения оптимального пути следования торговых агентов для приема заказов на продукцию в магазинах, супермаркетах и других торговых точках согласно определенного расписания, где указано, как часто следует посещать данную торговую точку.

Выполнен анализ предметной области и выбран наиболее эффективный алгоритм и технология, на которой будет построена система.

Сделан вывод о целесообразности ориентирования системы на веб-доступ и использование общедоступных картографических сервисов.

Общая постановка задачи

На текущий момент существует много систем, решающих сходные задачи, в основном они ориентированы на сектор логистики перевозок. Однако большая часть взаимодействует с локальными базами карт местности, что затрудняет масштабирование и регулярное обновление данных карт. Таким образом, становится очевидной необходимость в дальнейших разработках и исследовании в данной области научной деятельности.

В данной статье подробно рассмотрены возможные варианты решения задачи, выбран и обоснован наиболее эффективный подход. Рассмотрены тенденции развития современных приложений, выявлены их преимущества и недостатки в контексте решаемой задачи.

Постановка задач исследования

Для составления более подробной представления о существующие системах, рассмотрим две технологии построения систем оптимизации движения:

- веб-базированные системы оптимизации движения;
- настольные системы оптимизации движения.

Рассмотрим детали функционирования веб-базированных систем на примере картографических веб-сервисов Google (Google API).

Карта состоит из одного или нескольких наложенных друг на друга слоев, а каждый слой разбит на множество квадратных участков - т. н. «Тайлов». При отображении карты все тайлы сочетаются, слои накладываются друг на друга и формируется единое изображение. Формирование единого изображения производится автоматически и этот процесс скрыт от пользователя. На рисунке 1 изображен вид карты, разбитой на тайлы в разных масштабах.

Карта поддерживает работу со следующими типами координатных систем:

- Географическая система координат (используется по умолчанию);
- Декартова система координат;
- Пиксельные системы координат;
- Тайловая система координат.

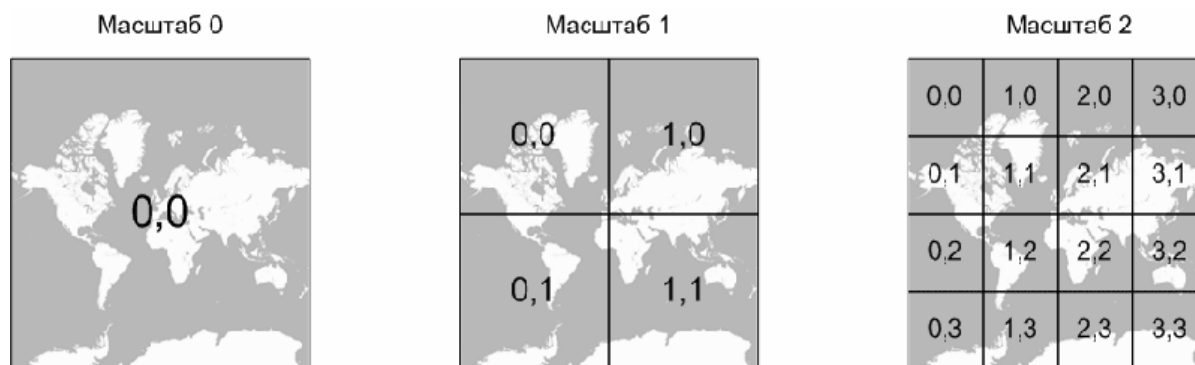


Рисунок 1 – Карта, разбитая на тайлы в разных масштабах

Земля имеет форму геоида, практически шара. Для указания местоположения точки на земной поверхности чаще всего применяются географические (геодезические) системы координат. Координаты точки в таких системах представляются тремя величинами: долготой, широтой и высотой.

Эти карты используют Всемирную геодезическую систему WGS-84 для адресации точек поверхности земного шара. Преобразование долготы и широты в координаты точек на плоскости называется картографической проекцией. Для отображения поверхности земли на плоскость в картах используется проекция Меркатора. Такое преобразование выполняется при преобразовании из географических в декартовы, пиксельные и тайловые координаты.

Рассмотрим архитектуру функционирования настольных систем оптимизации движения. Настольные системы базируются на другой технологии представления карт - векторных картах, поэтому сначала я определю что такое векторная графика.

Векторная графика - способ представления сложных объектов и изображений в компьютерной графике основан на использовании геометрических примитивов, таких как точки, линии, сплайны и многоугольники. Термин используется в противоположность к растровой графике, которая представляет изображения как матрицу фиксированного размера, состоящая из точек (пикселей) со своими параметрами.

Объекты на векторной карте хранятся не так, как на растровых картах: объекты хранятся в виде сплайнов, окружностей и других геометрических фигур, в отличие от набора координат и изображений местности в растровых картах.

Векторные карты имеют некоторые преимущества перед растровыми:

- размер, занимаемой описательной частью, не зависит от реальной величины объекта, что позволяет, используя минимальное количество информации, описать как угодно большой объект файлом минимального размера;
- информация об объекте сохраняется в описательной форме, можно бесконечно увеличить графический примитив, например, дугу окружности, и она останется гладкой. С другой стороны, если кривая представлена в виде ломаной линии, увеличение покажет, что она в действительности не кривая;
- при увеличении или уменьшении объектов толщина линий может быть задана постоянной величиной, независимо от реального контура.

Недостатками являются сложность хранения географической информации и трудности при описании сложных геометрических фигур, в отличие от растровых карт, которые сохраняют эту информацию в виде картинок.

В нашем случае можно пренебречь достоинствами векторной графики в пользу более гибких, простых в дополнении и использовании – растровых карт.

Современные приложения в основном ориентируются на веб-доступ, что делает разработку настольных приложений неактуальной. Преимуществами их использования является также несравнимо более широкий масштаб информационного покрытия. Рассмотрев данный вопрос необходимо определить, какая технология позволит в лучшей степени пользоваться преимуществами разрабатываемой системы.

Выбор технологии является определяющим при построении любой веб-системы, поэтому следует рассмотреть, какие средства и технологии доступны сегодня для архитекторов современных систем.

RIA (Rich Internet application, "богатое Интернет-приложение") - это программа, доступная через Интернет, богатая функциональностью традиционных настольных приложений, не поддерживаемой браузерами непосредственно. Как правило, приложение RIA:

- передает веб-клиенту необходимую часть пользовательского интерфейса, оставляя большую часть данных (ресурсы программы, данные и т.п.) на сервере;
- запускается в браузере и не требует дополнительной установки ПО;
- запускается локально в среде безопасности, называемой "песочница".

На рис. 2 с помощью диаграммы Венна демонстрируется место Rich Internet Applications среди других технологий программных систем [1].



Рисунок 2 –Диаграмма Венна

Работа традиционных веб-приложений сконцентрирована вокруг клиент-серверной архитектуры с тонким клиентом. Такой клиент переносит все задачи по обработке информации на сервер, а сам используется только для отображения статического контента (например, HTML-кода). Основной недостаток этого подхода в том, что все взаимодействие с приложением должен обрабатываться сервером, что требует постоянной отправки данных на него, ожидания ответа сервера и загрузки страницы обратно в браузер. При использовании технологии запуска приложений на стороне клиента, RIA могут обойти этот медленный цикл синхронизации за счет большего взаимодействия с пользователем. Эта разница примерно аналогична разнице между архитектурой с "тонким клиентом" (Thin client) и архитектурой с "толстым клиентом" (Fat client), а также между терминалом и мейнфреймом.

Рассмотрим архитектурные особенности построения RIA-приложений.

Сравнение принципов работы веб-приложения и RIA-приложения представлен на рис. 3 [2].



Рисунок 3 – Сравнение принципов работы традиционного веб-приложения и RIA-приложения

В таблице 1 приведены сравнительные данные профилирования традиционного веб-приложения (на примере сайта mail.ru) и RIA-приложения (на примере сайта gmail.com) [2].

Таблица 1. Сравнение традиционного веб-приложения и RIA-приложения.

Действие	Mail.ru	Gmail.com
Начальная страница	116kb 24 запроса	33 запроса 379 kb
Открыть письмо	17 запросов 71 kb	3 запроса 2 kb

Таким образом, можно формализовать задачу, можно выделить следующие этапы решения:

- создание серверного хранилища данных;
- создание серверной части системы, которая будет посылать и принимать данные по запросам клиентов (основная логика);
- реализация клиентской логики (логика на стороне клиента);
- создание интерфейса (уровень представления).

Рассмотрим каждый из этапов в отдельности.

Серверное хранилище данных будет содержать географические данные, информацию, которую необходимо накапливать для поиска кратчайших и альтернативных путей передвижения агента.

Серверная часть – основная логика приложения, которая отвечает за взаимодействие с клиентом. Здесь производится реализация выбранного алгоритма, который, являясь сердцем системы, совмещает географические данные из хранилища с расписанием посещения торговых точек, поставляемым с помощью веб – сервиса.

Клиентская часть – отвечает за получение и отображение на интерфейсе географических данных, а также передачу на сервер географических данных из картографического источника.

Выводы

Рассмотрев все существующие веб - технологии, можно увидеть, что у разработчика есть широкий выбор для построения системы в зависимости от ее целей. В случае системы, которая должна быть интегрирована с картографическим сервисом, выбор ограничивается теми средствами, которые поддержат этот сервис.

На сегодня можно выделить лишь 2 поставщики картографической информации, имеющие достаточно обширную базу данных в восточном регионе:

- Сервис карт Google;
- Сервис карт Yandex.

Работа с ними может быть организована с помощью двух подходов:

- Использование javascript API;
- Использование Flash API.

Таким образом из рассмотрения выпадают такие технологии как:

- WPF;
- Silverlight 3.0.

Следующим фактором в выборе технологии является интеграция с существующими средствами анализа корпоративных данных, ведь система будет частью корпоративной системы, поэтому должна совпадать с корпоративными стандартами. В данном случае - система ориентирована на платформу .NET, поэтому использование технологий фирмы Adobe - не возможно.

В таком случае выбор падает на использование Javascript API в связке с технологией ASP.NET используя Ajax, как транспорт для построения RIA, ведь приложение должно отвечать современным стандартам и требованиям скоростной обработки данных.

Рассмотрев возможные алгоритмы разрешения задачи, сравнив эффективность, результативность и временные затраты, необходимые для получения результата, оптимальным применимо к задаче является алгоритм А*.

Литература

- [1] Левашов А.А. Опыт применения RIA в бизнесе (на практических примерах). - Второе издание, изд.дом «Вильямс» Москва, С. Петербург, Киев, - 2009. 409 с.
- [2] Ковалев К.К. RIA - значит свобода – С. Петербург, - 2009. 150 с.
- [3] Майника Э. - Алгоритмы оптимизации на сетях и графах – изд.дом "Мир", Москва, - 1981. 503с.
- [4] ГОСТ 7.32-81. Отчет о научно-исследовательской работе. Общие требования и правила оформления.
- [5] ГОСТ 19.002-80 . Единая система программной документации. Схемы алгоритмов и программ. Правила выполнения.
- [6] ГОСТ 19.003-80. Единая система программной документации. Схемы алгоритмов и программ. Обозначения условные графические.