

## РАЗРАБОТКА ВЫСОКОТОЧНОЙ СИСТЕМЫ ГЕОПОЗИЦИОНИРОВАНИЯ ДЛЯ НЕБОЛЬШИХ НАСЕЛЕННЫХ ПУНКТОВ

И. А. Ботыгин, А. С. Вишневецкий

Институт кибернетики Национального исследовательского  
Томского политехнического университета, 634034, Томск, Россия

---

УДК 004.4:004.7

Представлена разработка программного обеспечения системы геопозиционирования для небольших городов и населенных пунктов, позволяющего определить положение пользователя с минимально допустимой погрешностью.

**Ключевые слова:** программное обеспечение, геолокация, геопозиционирование.

Presented the development of software systems geolocations small cities and towns, which allows to determine the position of the user with the minimum allowable error.

**Key words:** software, geolocation, geoposition.

В настоящее время достаточно широко распространены геолокационные сервисы. Геолокация, точнее, геопозиционирование дополняет многие web-сервисы и особенно актуально для таких серверных приложений, которые доступны при использовании современных мобильных телефонов. Это очевидно, так как геопозиционирование – это не только определение реального географического положения и мониторинг динамики передвижений, но и дополненная виртуальная реальность, геолокационные социальные сети, мобильный маркетинг и др.

Вместе с тем при определении географических координат объектов возникают проблемы точности, достоверности и конфиденциальности полученной информации. В населенных пунктах и небольших городах, где современные коммуникационные технологии не получили достаточного развития, особенно остро стоит проблема высокоточного геопозиционирования. Представляет интерес разработка приложения, позволяющего определить с минимально допустимой погрешностью положение пользователя в таких районах. В настоящее время можно выделить две основные системы геопозиционного определения координат: 1) спецификация HTML5 Geolocation; 2) недавно анонсированный сервис Яндекс.Локатор (для России). Данные системы предоставляют возможность определять координаты объектов с помощью не только мобильного устройства, но и персонального компьютера.

Широко распространенной коммуникационной технологией для персональных компьютеров является сеть Wi-Fi. Разработана концепция определения координат с помощью данных сетей. Алгоритм определения данных, полученных от Wi-Fi, требует наличия большого количест-

ва ресурсов и заключается в следующем. Каждый пользователь, находясь в зоне действия Wi-Fi-сетей, имеет возможность получить персональный адрес каждого Wi-Fi-маршрутизатора, называемый MAC-адресом [1]. Однако проблема состоит в том, что у маршрутизатора как такового отсутствуют координаты. Разработчики систем HTML5 Geolocation и Яндекс.Локатор выбрали разные концепции. Компания Google, которой принадлежат права на спецификацию HTML5 Geolocation, встраивает специальную службу в их разработку (систему Google Android), которая при обнаружении новой сети Wi-Fi и подключении к сети GPS сохраняет текущие координаты в долговременную память устройства. В момент подключения устройства к сети Интернет все данные передаются на серверы Google в "прозрачном" для пользователя режиме [2]. В данном случае компания Google нарушает политику конфиденциальности, но за счет большого количества пользователей система практически всегда способна определить координаты с достаточно низкой погрешностью [3]. Разработчики Яндекс.Локатор используют другой способ. Они отправляют по городам России машины, которые записывают историю перемещений и через малые промежутки времени сканируют наличие Wi-Fi-сетей в радиусе перемещения [4]. Этот метод требует больших вложений, но при этом не нарушается политика конфиденциальности. Именно такая схема лежит в основе разработанной системы геопозиционирования.

Схема геолокации заключается в следующем. Если пользователи, обладающие устройством с GPS-приемником и доступом в Интернет, поставят себе специальное приложение, то можно получить достаточно точную координатную картину. Эта схема требует временных затрат, но включает преимущества рассмотренных выше систем геопозиционирования. В настоящей работе выполнена разработка инструментария (программный интерфейс API), позволяющего определить координаты объекта с более высокой точностью, чем при использовании рассмотренных выше вариантов.

**Обобщенная функциональная структура системы геопозиционирования.** Система для определения географических координат объекта имеет серверную и клиентскую части (рис. 1). Серверная часть предназначена для получения и анализа данных, переданных от клиента, и последующей выдачи их клиенту в необходимом формате. Клиентская часть служит для определения характеристик и текущих параметров GPS, Wi-Fi и других сетевых устройств. Иными словами, система построена по принципу клиент-серверной архитектуры. Каждый элемент этой архитектуры выполняет свою функцию: сервер владеет и распоряжается информационными ресурсами системы, клиент имеет возможность использовать их. Сервер базы данных представляет собой мультипользовательскую версию СУБД, параллельно обрабатывающую запросы, поступившие от всех клиентов. В число его задач входят реализация логики обработки транзакций с использованием необходимой техники синхронизации – поддержки протоколов блокирования ресурсов, обеспечение, предотвращение и (или) устранение тупиковых ситуаций.

В многоуровневой структуре интерфейса верхним звеном является клиентское приложение, выполняющее роль "сборщика" информации о координатах. В качестве клиента в системе может использоваться любое современное мобильное устройство, а также персональный компьютер. В частном случае, без использования специализированного оборудования, таким "сборщиком" является мобильное устройство со встроенным чипом GPS и выходом в сеть Интернет.

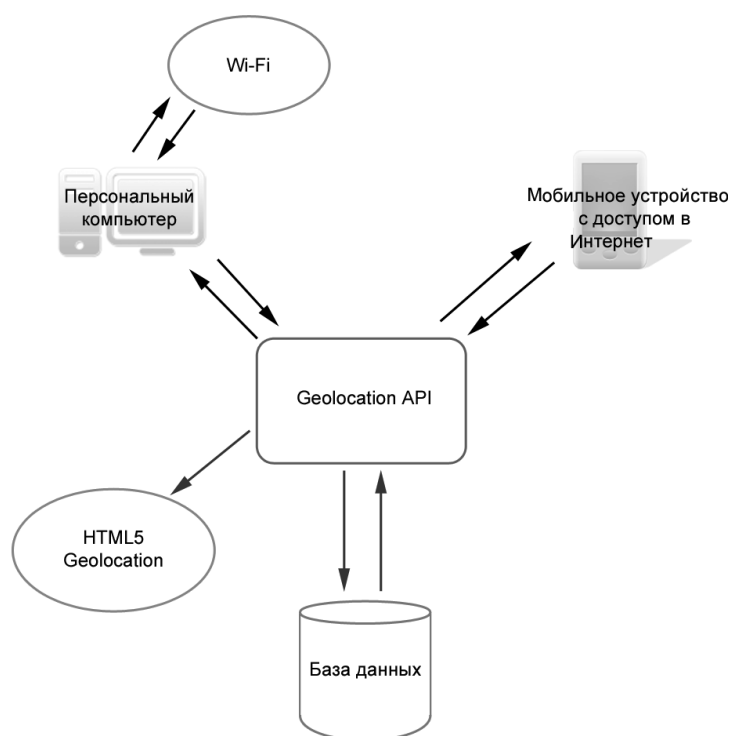


Рис. 1. Схема взаимодействия элементов системы геопозиционирования

**Основные функции серверной части.** Серверная часть служит для обработки данных, переданных от клиента. Среди таких данных могут быть:

- данные, полученные от GPS- и (или) ГЛОНАСС-чипов;
- информация о доступных точках Wi-Fi;
- информация о сетевом соединении (ip-адрес, MAC-адрес, шлюз).

Серверной частью выполняются все основные скрипты, а также осуществляется обращение к сторонним сервисам для сбора и дальнейшего анализа полученных данных. Среди основных функций серверной части можно выделить следующие:

- обработка GET- и POST-запросов от различных устройств;
- выполнение запросов к базе данных;
- доступ к классам определения координат;
- доступ к сторонним средствам определения координат;
- программный интерфейс для сторонних разработчиков;
- обработка ошибок.

В центре системы находится основной класс, который занимается обработкой всех поступающих извне запросов. Данный класс должен осуществлять запросы к базе данных, для чего используется специальный класс для работы с базой данных. Основная часть имеет доступ к классам TriangleClass, WifiClass, IpClass, которые занимаются обработкой соответствующих данных. IpClass имеет доступ к классу для работы с базой данных, которая обновляется с помощью стороннего сервиса. Основной класс обменивается данными с сервисом HTML5 Geolocation для получения координат, если ни одному встроенному классу не удалось определить координаты.

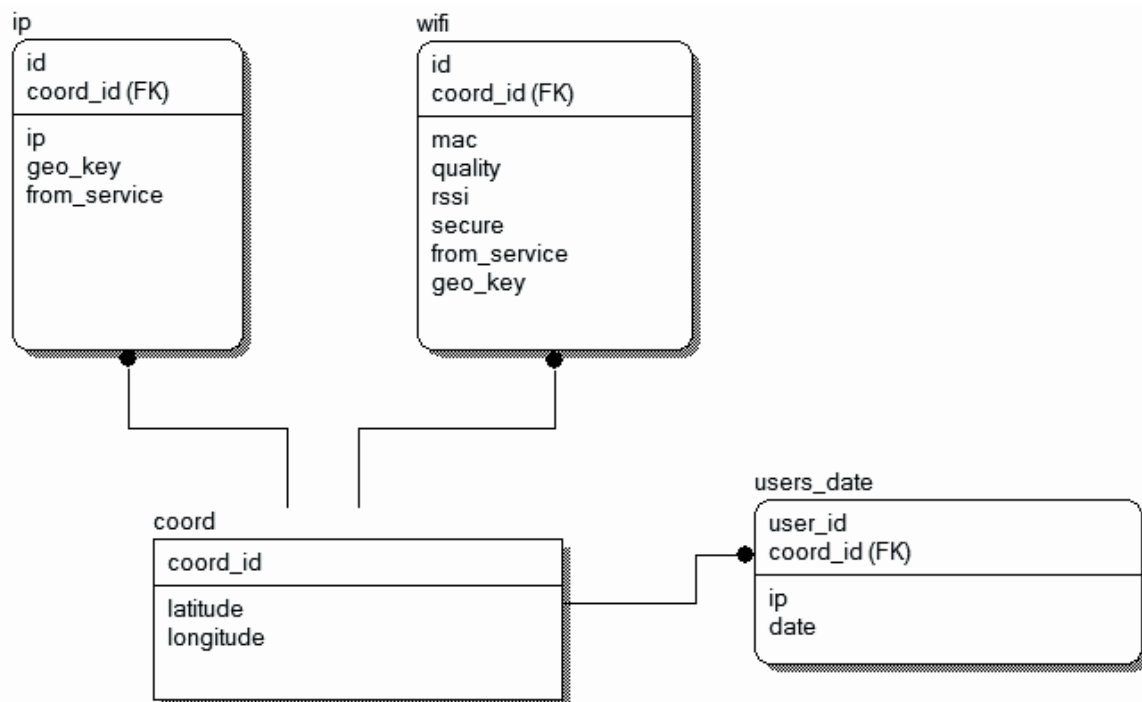


Рис. 2. Структура основной базы данных системы

**Основные функции клиентской части.** Клиентская часть занимается сбором и передачей данных серверу, а также приемом и выводом координат пользователю. Основные функции выполняются с помощью сторонних библиотек и API-сервисов. Предоставляемые библиотеками и API данные не влияют на точность определения координат. В данной работе представлены приложения для персонального компьютера и мобильного устройства на базе операционной системы AndroidOS.

Различие приложений заключается в различном количестве устройств, к которым данное устройство может получить доступ. Для персонального компьютера недоступно определение координат с помощью GPS-чипа, так как чип встраивается в отдельное устройство и для каждого такого устройства необходимо разрабатывать отдельную программу вследствие различия драйверов. Мобильное устройство имеет доступ ко всем возможным определяемым системой типам устройств, в том числе к GPS, что позволяет наиболее точно определить координаты.

Клиентская часть для персонального компьютера использует основные методы библиотеки Wi-Fi Manager Library стороннего разработчика для получения данных от Wi-Fi-устройств. Для получения параметров сетевых устройств используются функции, встроенные в операционные системы типа Windows.

Клиентская часть для мобильного устройства на базе операционной системы AndroidOS использует встроенные в эту систему библиотеки для доступа к устройствам и обмена информацией с ними. Следовательно, отсутствует необходимость использовать сторонние библиотеки.

**Структура базы данных системы.** Система использует основную базу данных для хранения и выборки информации о положении объекта и базу данных GeoIP для определения города в вынужденных случаях. Структура основной базы данных представлена на рис. 2.

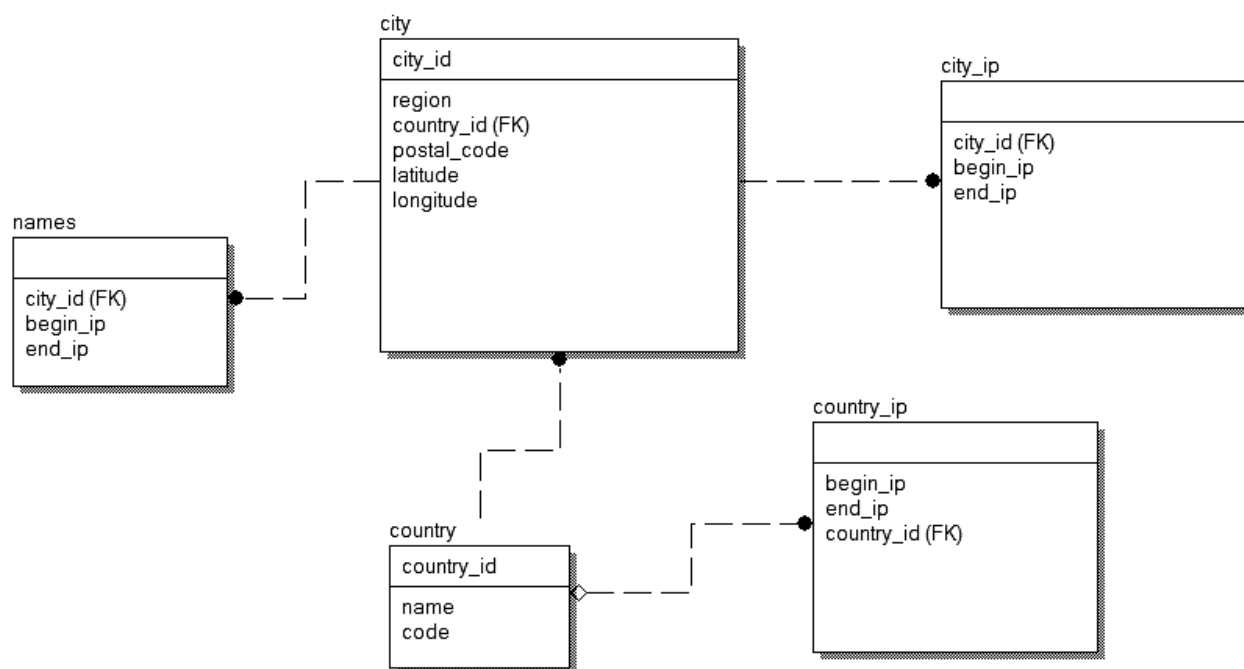


Рис. 3. Структура базы данных GeoIP

База данных содержит четыре таблицы для хранения всех координат системы и истории обращения пользователей к базе данных. Таблица coord служит для хранения широты и долготы. Данная таблица связана типом связи Один-Ко-Многим с таблицами ip, wifi, users\_date.

В таблице ip хранятся ip-адреса объектов, которые определяются на стороне сервера. Если ip-адрес определен, он заносится скриптом в базу данных. Столбец geo\_key служит для хранения ключа, который необходим для того, чтобы получить информацию от сервиса HTML5 Geolocation. Такая возможность возникает в случае, если в базе недостаточно данных для определения координат пользователя. Информация, полученная от HTML5 Geolocation, в дальнейшем заносится в базу данных и кешируется.

Таблица wifi предназначена для хранения следующих данных о wi-fi-соединении: mac-адрес, качество сигнала, rssi (мощность сигнала), защищенность канала. Столбец geo\_key служит для хранения ключа, который необходим для того, чтобы получить информацию от сервиса HTML5 Geolocation.

Таблица users\_date служит для хранения времени обращения клиента к серверу. В столбце date, имеющем тип DATETIME, хранятся дата и время обращения. Клиент определяется по ip-адресу удаленного соединения.

В случае невозможности определения координат основным классом системы и сервисом HTML5 Geolocation система обращается к сервису GeoIP с помощью соответствующего класса. Структура базы данных GeoIP представлена на рис. 3.

База данных GeoIP поддерживает иерархию географии страна – город, которая хранится в нормализованном виде в СУБД. Данные о диапазонах IP-адресов со связями ко всем элементам географии содержатся в таблицах names, city\_ip, country\_ip. Таблицы country и city служат для хранения информации о странах и городах соответственно. Данные о диапазонах ip-адресов бе-

рутся с сайта и обновляются приблизительно один раз в месяц. Обновление данных с сайта не рассматривается, так как данные берутся со стороннего сервиса.

Для защиты серверной части от SQL-инъекций и других серверных уязвимостей используется фреймворк ZendFramework, который выполняет роль “обертки” безопасных функций.

**Заключение.** Таким образом, основным компонентом системы геопозиционирования является серверное приложение, осуществляющее регистрацию объектов и предоставляющее геолокационную информацию (координаты объекта и точность определения) в удобном для клиента виде. В соответствии с приоритетами точности методов определения координат, сервер осуществляет поиск в основной базе данных, полученных от клиента. В случае невозможности определить координаты в основной базе данных разработанный сервис получает доступ к системам сторонних разработчиков и передает им полученные от клиента данные в соответствующем формате. Обработка исключительных ситуаций осуществляется за счет встроенного в серверную часть специального обработчика ошибок.

Основными особенностями системы являются высокая точность определения координат (в данный момент в городских масштабах), возможность масштабирования с помощью объектно-ориентированного подхода, различные способы вывода информации (форматы HTML, XML и JSON), а также доступ к известным сервисам (база ip-адресов GeoIP, HTML5 Geolocation, Яндекс.Локатор) в случае невозможности определения координат.

#### Список литературы

1. Беспроводные сети Wi-Fi на службе бизнеса // Shark ID. [2012]. [Electron. resource]. <http://www.shark.ru/catalog/index.php?ID=159>.
2. GEOLOCATION API Specification // World Wide Web Consortium. [2012]. [Electron. resource]. <http://dev.w3.org/geo/api/spec-source.html>.
3. Безопасность HTML5 – часть четвертая // SecurityLab. [2012]. [Electron. resource]. <http://www.securitylab.ru/analytics/419889.php>.
4. API Яндекс.Детектор// Яндекс. [2008-2012]. <http://api.yandex.ru/detector/>.

*Ботыгин Игорь Александрович – канд. техн. наук, доц. Института кибернетики  
Томского политехнического университета; тел.: (382-2) 42-04-05; e-mail: bia@tpu.ru;*

*Вишневский Алексей Сергеевич – студент Института кибернетики  
Томского политехнического университета; e-mail: mango.tomsk@gmail.com*

Дата поступления – 05.10.12 г.