



Магистерская программа «Программное обеспечение вычислительных  
сетей»

Магистерская диссертация

# ГЕО-КОНТЕКСТНЫЕ ПРИЛОЖЕНИЯ НА WINDOWS PHONE

Работу выполнил:

**Юсуф-заде Гулам Эльчин оглы**

Научный руководитель:

*с.н.с. к.ф.-м.н.* **Намиот Дмитрий Евгеньевич**

*Подпись научного руководителя:*

## Оглавление

|       |  |    |
|-------|--|----|
| 1     | Аннотация .....  | 4  |
| 2     | Введение .....   | 5  |
| 3     | Постановка задачи .....                                    | 7  |
| 4     | Обзор существующих решений .....                           | 8  |
| 4.1   | Цели и критерии сравнения .....                            | 8  |
| 4.2   | gMaps.....   | 9  |
| 4.3   | NokiaFriendsView .....                                     | 11 |
| 4.4   | Social recommendation system .....                         | 13 |
| 4.5   | Google Now .....   | 17 |
| 4.6   | Выводы.....  | 19 |
| 5     | Исследование и построение решения задачи .....             | 21 |
| 5.1   | Регистрация сервисов .....                                 | 22 |
| 5.2   | Унификация результатов .....                               | 24 |
| 5.3   | Определение географических координат .....                 | 26 |
| 5.3.1 | Определение координат на основе спутниковых данных .....   | 26 |
| 5.3.2 | Определение координат на основе данных сотовых сетей ..... | 28 |
| 5.3.3 | Выводы .....   | 29 |
| 5.4   | Краткие выводы .....                                       | 30 |
| 6     | Описание практической части.....                           | 31 |
| 6.1   | Инструментарий.....  | 31 |
| 6.2   | Декомпозиция задачи .....                                  | 33 |
| 6.3   | Сетевой компонент .....                                    | 34 |
| 6.4   | Клиентский компонент.....                                  | 36 |
| 6.4.1 | Шаблон проектирования.....                                 | 38 |
| 6.4.2 | Фоновое обновление и нотификации .....                     | 40 |
| 6.5   | Иллюстрация результатов работы.....                        | 43 |

|     |                                     |    |
|-----|-------------------------------------|----|
| 6.6 | Возможные направления развития..... | 44 |
| 7   | Заключение.....                     | 45 |
| 8   | Библиография.....                   | 46 |

## **1 Аннотация**

В данной работе изучается возможность применения универсального подхода к реализации приложений и сервисов, ориентированных на работу с гео-контекстными данными. Также объясняется понятие контекста и возможности его применения. Помимо этого приводится конкретная реализация прототипа сервиса, выполняющего работу по поиску и агрегации данных, находящихся в различных информационных источниках.

## 2 Введение

За последнее время произошел существенный рост производительности мобильных устройств (смартфонов, планшетных компьютеров и т.д.). На настоящий момент подобного рода системы демонстрируют скорость работы, приближающуюся к скорости персональных компьютеров. В связи с этим расширился и спектр задач, которые могут выполняться на мобильных системах. Одной из основных тенденций стало развитие так называемых контекстно-зависимых вычислений. Концепция контекстно-зависимых вычислений, применительно к мобильным устройствам, исходит из того, что результат работы многих приложений может быть значительно улучшен за счет использования контекста, под которым в данном случае могут подразумеваться такие параметры, как время суток, текущее географическое положение, сетевая активность пользователя и т.п. Все перечисленные составляющие могут отслеживаться в режиме реального времени, благодаря многочисленным датчикам и сенсорам, которыми оснащена значительная часть существующих портативных устройств.

Одними из первых определение контекста предложили в 1994 году Б. Шилит и М. Теймер. Контекст — это информация о местоположении, людях, находящихся рядом, и их изменениях. Ими же было дано определение контекстно-зависимых вычислений как программного обеспечения, использующего контекст. Позже К. Деэмбыло введено более общее определение контекста, как любой существенной информации, характеризующей ситуацию, в которой оказалась сущность. Под сущностью подразумевается человек, место или объект, считающийся важным для взаимодействия с пользователем приложения (конечным пользователем или же другими приложениями).

Контекстно-зависимые приложения, как правило, обладают следующими свойствами:

- Автоматизация. Например, автоматический вход в учетную запись пользователя при его приближении или автоматическая переадресация звонков с мобильного телефона на городской.

- Адаптация. Приложение может менять свое поведение в зависимости от доступного оборудования и особенностей среды, окружающей пользователя.
- Персонализация. Пользователь может изменить, специализировать поведение программы в определенном контексте.

Контекстно-зависимые вычисления ориентированы в первую очередь на конечного пользователя. Подобного рода системы динамически адаптируются к изменяющимся окружающим обстоятельствам и позволяют предоставить информацию, важную именно в конкретный момент **Ошибка! Источник ссылки не найден.**

Операционная система Microsoft Windows Phone является достаточно новой разработкой. В 2010 году были представлены первые устройства с данной операционной системой. Windows Phone заменила собой более раннюю Windows Mobile. На данный момент система находится в фазе своего активного развития. Вышло несколько обновлений системы, добавляющих новую функциональность и устраняющих ошибки системы. В 2012 году была представлена следующая версия платформы – Windows Phone 8. Именно эта редакция операционной системы предоставляет возможность объединить планшеты, смартфоны и персональные компьютеры в единую экосистему, то есть облегчает разработчикам перенос программного обеспечения между этими устройствами.

Поскольку операционная система является достаточно молодой, на данный момент не существует приложений и сервисов, позволяющих находить местоположение интересующих людей по их сетевой активности, исходя из данных различных источников.

### **3 Постановка задачи**

Целью данной работы является разработка универсальной архитектуры и создание прототипа сервиса для мобильной платформы Microsoft Windows Phone, позволяющей определять географическое местоположение интересующих людей (друзей, коллег) на основе их сетевой активности. Основным компонентом функционирования приложения является работа с контекстной информацией. В данном случае в качестве ведущей информационной единицы будет выступать географическое местоположение пользователя приложения, иными словами подразумевается работа с гео-контекстными данными.

В качестве источников данных о сетевой активности интересующих людей выступают различного рода информационные источники, такие как социальные сети, web-ресурсы и т.п.

Создание архитектуры подразумевает выделение общих абстрактных компонентов (слоев) и описание функциональности каждого слоя.

Создание сервиса в данном случае подразумевает создание двух компонентов: сервера данных - модуля, отвечающего за сбор и унификацию данных из источников, и клиента на мобильное устройство с установленной операционной системой Windows Phone, а также обеспечение взаимодействия между указанными модулями. В качестве дополнительного требования к клиентскому приложению выдвигается возможность работы в фоновом режиме, то есть поддержка актуальности данных при неактивном приложении.

## 4 Обзор существующих решений

### 4.1 Цели и критерии сравнения

В качестве критериев оценки существующих решений были выбраны следующие параметры:

1. Интегрируемость разнородных источников данных. В рамках разрабатываемой системы происходит интеграция данных из различных источников, имеющих различный формат представления. Важным является выделение данного аспекта, как одного из основополагающих, учитывая тот факт, что анализируется сетевая активность исследуемых объектов. Как следствие необходима возможность исследования всевозможных источников, в которых мог быть оставлен информационный след.
2. Общая архитектура системы. Рассматривается общая архитектура компонентов, включаемых в решение. Исследуется возможность создания гибкого решения в рамках ограниченности вычислительных ресурсов мобильного устройства.
3. Целевая операционная система. Рассматриваются операционные системы, на которые ориентированы решения. В рамках предлагаемой работы была выбрана операционная система WindowsPhone. Поэтому особым интересом пользуются системы, рассчитанные именно на данную платформу.



## 4.2 gMaps

Одним из наиболее популярных приложений на платформе WindowsPhone, посвященных контекстно-зависимому анализу подобного рода, является gMaps, разработанный Алексеем Страхом и Александром Сороколетовым.

Программа обладает следующим функциями:

- Геокодирование<sup>1</sup> и реверсивное геокодирование<sup>2</sup>
- Локальный поиск — возможность найти интересующие заведения, прочитать отзывы о найденном объекте.
- Прокладка маршрута
- Отображение пробок и остановок общественного транспорта.
- Возможность отметить собственное местоположение и сделать его доступным для других пользователей.

Особенностью приложения является то, что вместо стандартных для платформы карт Bing используются карты Google, которые на данный момент имеют лучшее качество покрытия. Это решение связано с дополнительными трудностями, поскольку имеющиеся программные примитивы приспособлены только для базовых карт и имеют не очень высокую скорость работы со сторонними ресурсами. Однако только благодаря тесной интеграции с сервисом карт Google, стало возможным реализовать функционал, недоступный стандартным приложениям.

Помимо этого был обнаружен ряд прочих технических трудностей. Невозможным оказалась реализация offline<sup>3</sup> режима работы. Следует заметить, что в большинстве случаев работа ведется с картами, загружаемыми со стороннего ресурса, что увеличивает объем получаемых данных. Режим, при котором приложение могло бы опираться на карты, уже сохраненные в памяти, мог бы уменьшить подобные накладные расходы. Однако режим крайне непрост в реализации, более того в большинстве случаев противоречит пользовательскому соглашению, заключаемому с провайдером сервиса.

---

<sup>1</sup> Сопоставление объектам и записям географических координат

<sup>2</sup> Поиск объектов и записей по географическим координатам

<sup>3</sup> Работа в режиме без подключения к сети интернет

Следует заметить, что платформа WindowsPhone в ранних версиях поддерживала многозадачность весьма ограниченно. И для работы приложения требовалось, чтобы оно было постоянно запущено и активно. В случае открытия другой программы, приложение прекращало свою активность (отслеживание местоположения).

Приложение работает по так называемой клиент-серверной модели, где в качестве клиента выступает приложение на мобильном устройстве, а в качестве сервера — картографический сервис GoogleMaps.

### 4.3 NokiaFriendsView

Подавляющее большинство современных мобильных телефонов имеет возможность выхода в интернет. В приложении FriendsView, разработанном в NokiaBetaLabs в 2008 году, выход в интернет используется для создания социальной сети, в которой были бы представлены все интересующие пользователя люди.

Программа позволяет пользователю отмечать свое местоположение, публиковать статусы и сообщения. Данные, предоставленные пользователем, таким образом, привязываются к карте и становятся доступными как с мобильного устройства, так и с помощью web-сервиса. Это позволяет оставаться на связи с интересующими людьми и знать, где они находятся.

Основа, на которой работает программа, — обновление сообщений и данных о местоположении, иными словами, комбинация сервиса коротких заметок и геосервиса. Пользователи могут загружать обновления относительно своего текущего местопребывания, а также отмечать места, куда собираются пойти в ближайшее время и оповестить об этом круг заинтересованных лиц (друзей, семью). Как и в любом сервисе микроблогинга, доступна возможность комментирования публикаций.

Подобные социальные сети, создающиеся посредством мобильных терминалов(телефонов) через общий промежуточный уровень (обычно web), называются мобильной социальной сетью. Такие социальные сети имеют большой интерес как для бизнеса, так и для обычных пользователей, поскольку предоставляют возможность взаимодействия в режиме реального времени. Помимо этого, сервис может быть использован для предоставления рекомендаций, поиска и персонализации на основе данных генерируемых пользователем.

Интересной особенностью данного программного обеспечения является именно функция рекомендаций друзей на основе контекстных данных. Похожий сервис уже реализован в социальных сетях, таких как Facebook, Google+ , Linkedin и обычно базируется на основе анализа профиля пользователя на предмет похожего или совпадающего содержания (совместные интересы, работа в одной и той же организации) и отношений friend-of-a-friend. Эти алгоритмы имеют ряд недостатков

и часто страдают от эффекта friendspam и commentspam. Например, TwitRank-алгоритм ранжирования, используемого в Twitter, основан на количестве подписчиков и количестве комментариев, что приводит к риску рекомендации в друзья человека, занимающегося спамом и не представляющим никакого интереса.

#### 4.4 Social recommendation system

Группой исследователей из RomaTreUniversity был предложен socialrecommendationsystem – программное обеспечение, предназначенное для платформы GoogleAndroid, основной задачей которого является рекомендация пользователю так называемых точек интереса (PointsOfInterest). Этими точками интереса могут являться различного рода заведения, встречи, увеселительные мероприятия и т. д. Подобный функционал уже реализован в некоторых приложениях. Как правило, в них используются данные из различных социальных сетей (Facebook, Foursquare). Однако в большинстве случаев единственным критерием рекомендации того или иного мероприятия является географическая близость пользователя к месту проведения, а такие параметры, как день недели, время, погода, транспортная загруженность, не учитываются. В данной работе программное обеспечение собирает данные о пользовательских предпочтениях, планах и, исходя из собранной информации, дает рекомендации.

Задачами этого проекта ставились:

- 1) Создание методологии определения пользовательских предпочтений, отличной от существующих систем, основанных только на географическом местоположении.
- 2) Использование постоянно растущего количества информации в социальных сетях, отзывов пользователей и локальных web-ресурсов.
- 3) Разработка процедуры определения контекста для использования при выдаче рекомендаций.

Данное приложение работает с картографическим сервисом GoogleMaps, хотя в качестве источников данных могут быть использованы и другие ресурсы (BingMaps, Yahoo!Mapsi пр.). В качестве информационных ресурсов предполагалось использовать не только глобальные источники, но и локальные web-ресурсы, поскольку нередко именно они обладают более полной информацией об интересующих объектах.

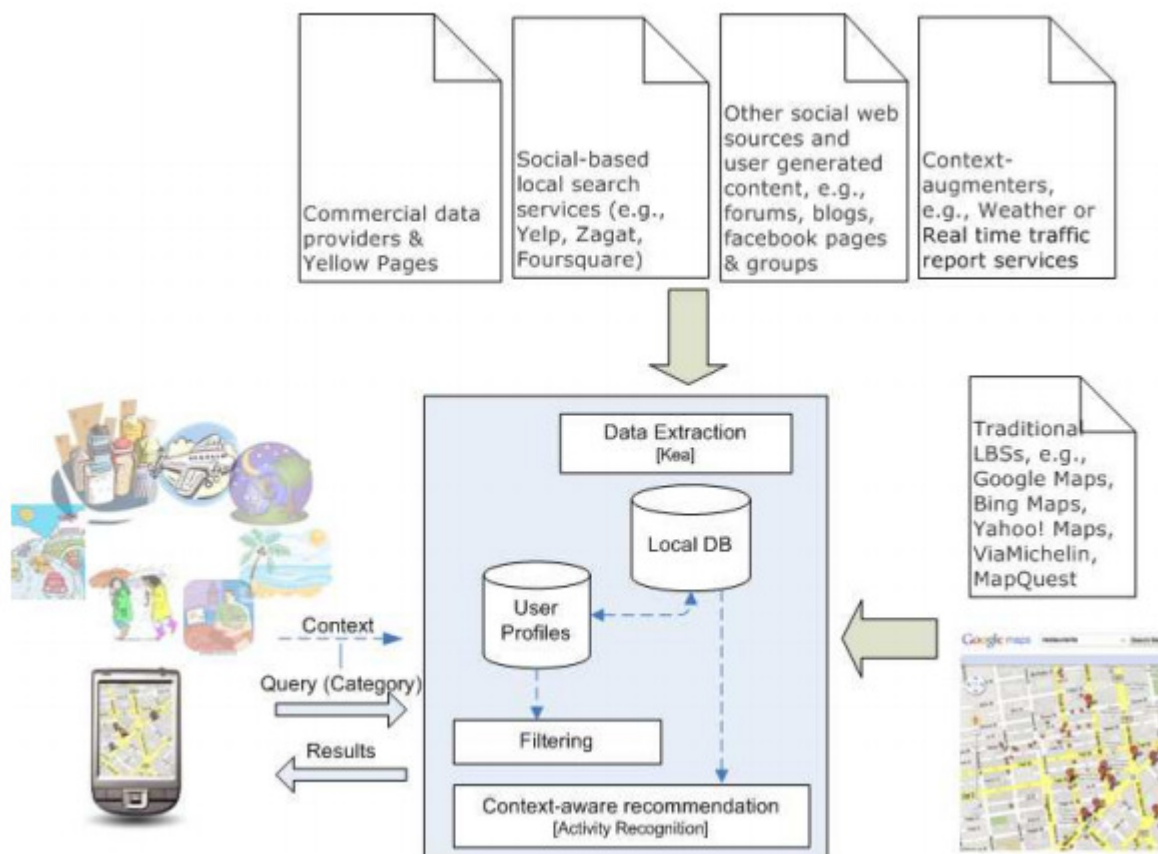
В качестве тестов был реализован ряд сценариев, учитывающих сколько времени пользователь готов затратить на посещение места, погоды, дорожной ситуации, и количество денежных средств, которые можно потратить. При этом, помимо основного предположения, программа предоставляла также ряд альтернатив.

Для решения поставленной задачи был разработан алгоритм Polar, который:

- моделирует пользовательские предпочтения с целью адаптировать рекомендации к специфическим нуждам;
- использует более детальные контекстные данные для уточнения результатов поиска;

Алгоритм работы приложения следующий: прежде всего, требуется аутентификация по паре ключ-пароль, после чего приложение получает доступ к профилю пользователя на web-ресурсе. Затем определяется текущее местоположение и демонстрируется карта, на которой присутствует отметка, обозначающая место, где сейчас находится устройство. После чего пользователю предоставляется возможность сделать интересующий запрос (строка текста). Обычно запросы характеризуют категорию точек интереса. Например, “пиццерия” или “fastfood” совпадает с названием категории, относящейся к предприятиям общественного питания. В случае несоответствия, например при поиске по ключевым словам “спагетти” или “французская кухня”, будут выбраны те точки интереса, чьи метаданные сходны с текстом запроса. Ранжирование найденных значений производится на основе пользовательских предпочтений и текущего контекста, после чего наиболее релевантные значения отмечаются на карте и сопровождаются подробным описанием. Приложение также дает возможность оценить точки интереса по шкале от 1 до 5 и поставить собственную пометку (может быть выбрана из предложенных системой или дана самим пользователем). Впоследствии полученная информация фиксируется в профиле и может быть использована для уточнения результатов поиска.

Приложение собирает данные об окружающем контексте и отправляет их вместе с запросом пользователя. Далее на web-сервисе происходит обработка. Следует заметить, что данные крайне персонифицированы и их утечка третьим сторонам недопустима. Поиск точек интереса по данной макрокатегории (класс точек интереса со сходными характеристиками: рестораны, бары, культурные события) тесно связано с текущими надобностями и предпочтениями пользователя.



Как показано на иллюстрации выше, пользователь взаимодействует с приложением, которое прикрепляет текущую контекстную информацию к запросу, после чего происходит обработка. Ключевым элементом при этом является локальная база данных с информацией из различных Web-ресурсов. Например, адреса и телефоны из телефонного справочника и более традиционные данные из картографических проектов. Результат поиска, как правило, представляет собой компиляцию данных из различных источников: блогов, новостных лент, форумов, социальных сетей. При этом могут демонстрироваться дополнительные подробности о месте, вроде

диапазона цен и телефона заказа столика. Вся эта информация хранится в локальной базе данных.

Система, ответственная за рекомендации, отслеживает текущий контекст и предоставляет персонализированный результат. Сама персонализация достигается в два шага. Пользователи ассоциированы с профилями, в которых представлены интересующие их макрокатегории. Сначала точки интереса, соответствующие данному контексту, изымаются из локальной базы данных. После чего результаты ранжируются, согласно пользовательским предпочтениям. Самые подходящие элементы дополнительно выделяются в пользовательском интерфейсе.



## 4.5 GoogleNow

Одной из наиболее крупных современных разработок в области контекстного анализа на мобильных платформах является программный продукт компании Google-GoogleNow<sup>4</sup>. Данное приложение представляет собой персональный ассистент пользователя, доступный на платформе GoogleAndroid.

GoogleNow является дополнением другого приложения — GoogleSearch, осуществляющего поиск в сети интернет. GoogleNow анализирует ежедневную повторяющуюся деятельность пользователя (нахождение в определенной географической точке или области, календарные события, поисковые запросы и т.п.) и на основе собранной статистики находит наиболее подходящую информацию. Данные потом формируются в специальный формат карточек

Программа активно использует систему GoogleKnowledgeGraph – систему, позволяющую уточнить поисковые запросы, исходя из анализа значений данных и их связей.

В целом, проект GoogleNow анализирует весьма большой спектр контекстных данных. На текущий момент приложение находит информацию о следующем:

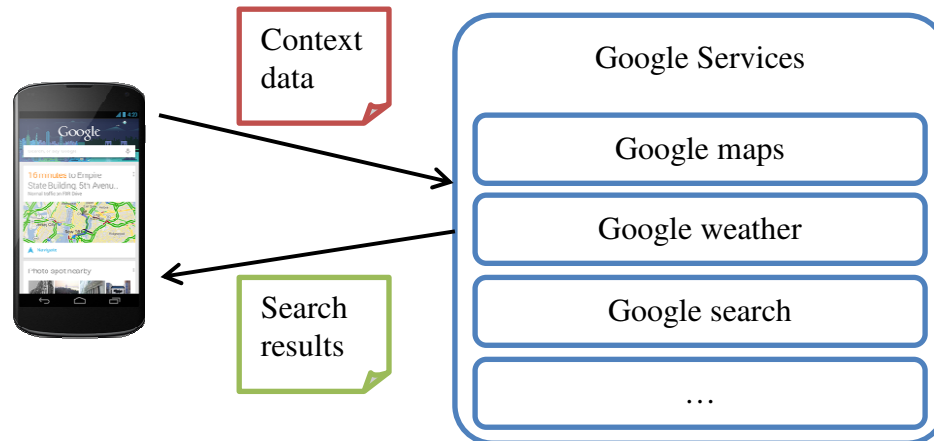
- Дни рождения
- События
- Данные о полетах
- Кинофильмы
- Новости
- Следующие встречи
- Спортивные результаты
- Трафик
- Точки интереса поблизости
- Погода
- Интересные точки для съемки рядом
- Биржевые сводки

---

<sup>4</sup><http://www.google.com/landing/now/>

- Дорога домой

Стоит заметить, что этот список возможностей зависит от географического региона и в некоторых странах может быть неполным.



GoogleNow использует также систему распознавания голоса GoogleVoice, которая позволяет осуществлять запросы на естественном языке. На данный момент система поддерживает все основные европейские языки, включая русский, а также значительное количество азиатских языков. Более того, количество языков постоянно увеличивается.

Проект построен согласно клиент-серверной архитектуре, где в роли клиента выступает непосредственно само приложение, а в роли информационных серверов – сервисы Google.

## 4.6 Выводы

Таким образом, рассмотрен ряд решений, ориентированных на работу с контекстными данными. Произведем сравнение результатов на основе критериев, выдвинутых ранее в главе «Цели и критерии сравнения».

| Название                     | Архитектура  | Целевая платформа | Интегрируемость разнородных ресурсов  |
|------------------------------|--|-------------------|---------------------------------------|
| gMaps                        | 2-хступенчатая. Клиент (мобильное приложение) – сервер (сервисы Google)                    | Windows Phone     | Только источники Google               |
| NokiaFriendsView             | 2-ступенчатая. Клиент (мобильно приложение)– сервер данных                                 | Symbian OS        | Только данные из источника агрегатора |
| Social recommendation system | 3-хступенчатая. Клиент (мобильное приложение) – сервер-агрегатор данных – источники данных | Google Android    | Есть                                  |
| GoogleNow                    | 2-хступенчатая. Клиент (мобильное приложение) – сервер (сервисы Google)                    | Google Android    | Только источники Google               |

В результате сравнения было выявлено, что только одна из рассматриваемых систем, использующих контекстные данные, позволяет интегрировать разнородные ресурсы внутри себя. Для этого в общей архитектуре решения присутствует дополнительная ступень, позволяющая взять на себя работу по сбору и унификации данных, а также работе с хранилищами данных.

Все прочие системы работают напрямую с сервисами, как правило, одного поставщика данных, и, как следствие, имеют возможность заранее предусмотреть формат принимаемых данных. Поэтому регистрация новых источников сопряжена с дополнительными трудностями.

Следствием исследования стало принятие трехступенчатой модели как основы архитектуры системы. Также подобная архитектура позволяет снять часть расчётов с клиентского устройства и эффективно реализовать поддержку и интеграцию разнородных ресурсов.

## **5 Исследование и построение решения задачи**

Для решения предложенной задачи была предложена определённая методология для создания и развития подобного рода сервисов в общем. Исходя из результатов исследований уже созданных систем, было решено провести декомпозицию задачи, выделить ее основополагающие концептуальные слои. В данной главе не рассматриваются технические подробности реализации, а только выдвигается концепция общей архитектуры. В рамках задачи выделяются четыре основных слоя:

1. Слой источников данных;
2. Слой предметных посредников;
3. Слой унификации;
4. Контекстный слой;

Каждый слой инкапсулирует строго определённую функциональность.

Слой источников данных представляет собой слой, в котором собраны необходимые источники, содержащие различные интересующие данные. Это могут быть различного вида Интернет-ресурсы, базы данных и т. п. Данный слой является внешним по отношению к системе. К нему осуществляются запросы на предмет извлечения необходимой информации.

Слой предметных посредников является слоем, содержащим все программные модули, ответственные за работу со слоем источников. Каждый из таких модулей, скрывает тонкости работы с каждым конкретным источником, при этом представляя удобный интерфейс для вышестоящего слоя.

Слой унификации. Этот слой оперирует данными, полученными со слоя предметных посредников. Результатом работы является сущность, содержащая запрошенную информацию в удобном формате. К каждой такой сущности применяется набор теоретико-множественных операций с целью конструирования наиболее точного результата.

Контекстный слой содержит в себе функциональность, требуемую для определения контекстных данных. Эти данные играют основополагающую роль в поиске

релевантной информации. Помимо этого информация используется в уточнении уже полученных данных.

## 5.1 Регистрация сервисов

Одним из основных моментов, связанных с поиском и анализом данных из различных источников, является задача унифицированного доступа к этим данным. Интересующий человек может быть зарегистрирован в целом ряде сетевых ресурсов. Каждый из подобного рода ресурсов имеет свой уникальный интерфейс общения. Иными словами, методы работы с каждым из подобных сервисов индивидуальны и, как правило, несовместимы друг с другом. Более того, некоторые источники могут и не иметь какого-либо инструментария для разработчиков сторонних решений. Следует также учитывать, что список источников может меняться динамически.

Естественным образом встает вопрос об интеграции всех подобных источников в рамках единой системы, а также унифицированный, единообразный доступ к интересующим ресурсам рассматриваемых сервисов.

В качестве решения этой задачи предлагается использовать адаптацию системы предметных посредников.

Посредник — это программный модуль, который использует информацию о некотором множестве или подмножестве данных для создания информации для пользователей более высокого уровня. В данном случае под пользователем или потребителем подразумевается другой программный элемент. Впервые подобная концепция была предложена Джо Вейдерхольдом — ученым из Стэнфордского университета.

Посредник инкапсулирует в себе всю специфику работу с конкретным источником данных. Он предоставляет унифицированный интерфейс для сущностей более высокого уровня и позволяет достичь единообразного обращения к интересующим ресурсам.

Таким образом, сущности более высокого уровня не знают об особенностях представления и формы хранения интересующей информации, будь то база данных, web-сервис или же что-то иное. К предметному посреднику осуществляется запрос в некотором стандартном виде, после чего этот запрос трансформируется в форму,

понятную конкретному ресурсу. Происходит выделение именно необходимой информации. В качестве параметров запроса могут выступать гео-контекстные данные, полученные с клиентского устройства.

Следует привести следующую классификацию подходов к интеграции информационных ресурсов.

- Процедурный подход к интеграции информации из заранее выбранных ресурсов называется GAVили GlobalasView. Интеграция информации из ресурсов при данном подходе основана на применении специальных процедур.В случае изменения информационных потребностей или ресурсов требуется повторная генерация посредника.
- Декларативный подход к интеграции информации из произвольных ресурсов по заранее определенным информационным нуждам называется LAV (LocalasView). Посредник содержит механизмы для изменения запросов согласно описанию ресурсов. Измененные запросы содержат в себе исходные.

В рамках данной системынаиболее подходящим выглядит подход LAV, однако и подход, основанный на GAV, также может быть реализован. Все зависит от особенностей каждой разворачиваемой информационной системы.

Следует заметить, что в пределах предметного посредника также происходит приведение полученных результатов запроса к некоторому единообразному виду, понятному для сущностей более высокого уровня.

Таким образом, функции предметного посредника в условиях данного подхода к решению задачи следующие:

1. Единообразный интерфейс доступа к данным;
2. Соккрытие деталей реализации запросов к конкретной системе;
3. Предоставление результатов в стандартизованном виде.

## 5.2 Унификация результатов

После получения результатов запроса от предметного посредника сущность более высокого уровня должна принять на себя обязанность по унификации. Результаты могут содержать разную степень детализации, однако вид их представления, исходя из требований к посреднику, будет одинаковым.

Задачей компонента унификации является конструирование некоторого универсального представления, составленного из результатов. В некоторых случаях, если количество информационных ресурсов фиксировано и достаточно мало, функции по конструированию конечного результата могут быть переложены на сами предметные посредники, однако это повлечет за собой увеличение сложности соответствующего модуля и потерю гибкости решения.

Для того чтобы эффективно выполнять конструирование, необходимо определить ряд операций над результатами. Эти операции могут быть реализованы как фильтры результатов, полученных от предметного посредника. В общем случае полученный результат можно рассматривать как множество пар ключ-значение.

Определим следующие операции:

- Пересечение. Пересечением двух множеств будем называть множество, состоящее из элементов, которые содержатся как в первом множестве, так и во втором. В более строгой математической форме эту операцию можно записать следующим образом.

$$\forall A, B: A \cap B = \{x | x \in A \wedge x \in B\}$$

- Объединение. Объединением двух множеств будем называть множество, состоящее из элементов, которые содержатся хотя бы в одном из множеств.

$$\forall A, B: A \cup B = \{x | x \in A \vee x \in B\}$$

- Разность. Разностью двух множеств будем называть множество, состоящее из элементов принадлежащих одному множеству, но не принадлежащих другому.

$$\forall A, B: A \setminus B = \{x | x \in A \wedge x \notin B\}$$



Таким образом, применяя композицию этих операций, можно создать представление, включающее в себя необходимую информацию из соответствующих источников.

Следует заметить, что конечный пользователь может значительно уменьшить данные сконструированного результата путем применения дополнительных фильтров.

Приведем пример работы модуля унификации. Предположим, что результатом работы предметных посредников стали следующие множества:

Результат 1

Firstname: John

Lastname: Dow

LastPlaceVisited: NewPark

Результат 2

Firstname: John

Lastname: Dow

PhotoId: 123

После применения операции объединения и изъятия дубликатов будет сконструировано следующее множество значений.

Firstname: John

Lastname: Dow

LastPlaceVisited: NewPark

PhotoId: 123

Важно заметить, что чем больше источников будет обработано, тем более детализированный результат можно получить.

### 5.3 Определение географических координат

Рассмотрим еще один важный аспект системы – определение географического местоположения пользователя. Координаты пользователя позволяют конкретизировать и уточнить данные, полученные на этапе поиска данных. Например, могут быть отброшены данные о людях, находящихся слишком далеко. Определение координат происходит на клиентском устройстве и впоследствии включается в запрос к системе поиска данных.

Рассмотрим основные способы определения географических координат.

#### 5.3.1 Определение координат на основе спутниковых данных

Наиболее точным методом определения координат является метод с использованием данных спутниковых навигационных систем. В данный момент на орбите Земли развернуты две спутниковые группировки, позволяющие определять координаты объектов на поверхности планеты. Это система GPS (США) и система ГЛОНАСС (Россия), также в активной фазе развития находится европейский проект спутниковой навигационной системы – Galileo (финал намечен на 2020 год). Спутниковые навигационные системы состоят из трех основных сегментов:

- 1 Космического сегмента;
- 2 Наземного управляющего сегмента (комплекса управления);
- 3 Абонентских приемников (пользовательского сегмента).



Рисунок 1: Упрощенная схема функционирования GPS

Космический сегмент состоит из спутников находящихся на орбите. Каждый из спутников передает сигнал, который и используется для определения местоположения. Стоит заметить, что приемники сигнала не имеют возможности отправить на спутник какой-либо сигнал. Таким образом, подобная система является пассивной.

Наземный управляющий центр занимается контролем спутниковой группировки, определением положения каждого из спутников, мониторингом состояния и коррекцией.

Абонентский приемник принимает сигнал со спутников и по времени прохождения сигнала с высокой точностью производит расчет географической позиции. Обычно для этого требуется 4 спутника, причем именно для определения координат используются три, а четвертый используется для синхронизации часов. Средняя погрешность определения местоположения 4-10 метров. К недостаткам этого метода относится ненадежность приема сигнала. В условиях плотной городской застройки или же при облачной погоде, сигнал может быть утерян или же значительно ослаблен. Стоит также заметить, что в помещениях данный метод не работает, поскольку сигнал теряется из-за стен и перекрытий.

### 5.3.2 Определение координат на основе данных сотовых сетей

Информация о местоположении может быть получена как от спутниковых систем навигации, так и от существующей (или расширенной измерительными модулями) инфраструктуры сети сотовой связи.

Выделяют следующие категории технологий определения местоположения в сотовых сетях:

- На основе мобильной станции (MS-based): мобильное устройство (в терминологии MS – MobileStation, мобильная станция) измеряет уровень сигнала и определяет местоположение независимо от сотовой сети. К числу технологий определения местоположения на базе MS относится измерение координат с помощью спутниковых навигационных систем (GPS, ГЛОНАСС) или технология A-GPS.
- С поддержкой данных сети (Network-assisted): процесс определения местоположения происходит главным образом в MS, однако MS получает поддержку от сети. Примером может служить технология A-GPS, в том случае, когда от эталонного навигационного приемника на терминал через внешнюю наземную сеть поступают уточненные данные, позволяющие терминалу значительно повысить результаты измерений от созвездия спутников GPS. Другим примером служит технология определения наблюдаемой разности времени приема сигналов OTDoA (Observed Time Difference of Arrival): сеть передает мобильной станции необходимую информацию о задержках сигналов для вычисления данных местоположения.
- С поддержкой мобильной станции (MS-assisted): MS измеряет уровни принимаемых от базовой станции (BTS – Base Transceiver Station) сигналов, а сеть определяет местоположение, например, с помощью усовершенствованной технологии измерения наблюдаемой разности времени приема сигналов E-OTD (Enhanced Observed Time Difference) для систем GSM или указанной технологии OTDoA для UMTS.
- На базе сети (Network-based): сеть выполняет все функции, связанные с определением местоположения. Наиболее известным примером является

технология идентификации соты (Cell ID), для которой не требуется определение параметров сигналов, так как это является внутренним свойством сетей мобильной связи. Точность определения местоположения уменьшается с увеличением радиуса сот с нескольких сотен метров до нескольких километров.

Другим примером технологий определения местоположения на базе сети является технология определения времени поступления сигналов (Time of Arrival, ToA), согласно которой на трех базовых станциях измеряется время поступления сигналов доступа от одной мобильной станции. Имеется множество способов определения

местоположения мобильных станций, многие из которых позволяют определить местоположение абонента с высокой точностью.

В общем случае, система, основанная на данных инфраструктуры сотовых сетей, имеет меньшую точность относительно спутниковых систем. Тем не менее, она может применяться там, где прием сигнала спутниковой системы будет затруднен.

### *5.3.3 Выводы*

В дополнение к сказанному ранее, следует заметить, что чаще всего используется комбинация перечисленных методов. В общем случае, когда требуется высокая точность, используется спутниковая навигационная система. В данный момент не существует возможности более точного определения географических координат. Если же прием сигнала затруднен или невозможен, то местоположение может быть с меньшей точностью уточнено с помощью инфраструктуры сотовой сети.

В целом, можно заключить, что современные методы определения координат обладают достаточной точностью, чтобы определить местоположение объекта вплоть до улицы или конкретного здания. Эти данные впоследствии, как было сказано ранее, могут быть использованы для уточнения результатов поиска.

## 5.4 Краткие выводы

Перечисленные в предыдущих главах компоненты являются основополагающим для создания системы, анализирующей сетевую активность интересующих людей. Благодаря ей, в системе может быть зарегистрировано произвольное количество источников данных. Более того, полученная информация может быть сгруппирована и дополнена путем использования данных из различных источников.

## 6 Описание практической части

### 6.1 Инструментарий

В качестве среды разработки для приложения была выбрана Microsoft Visual Studio, поскольку на данный момент это единственный программный комплекс, позволяющий создавать приложения, ориентированные на платформу Windows Phone. Данная интегрированная среда разработки позволяет разрабатывать консольные приложения, приложения с графическим интерфейсом, на основе технологии Windows Forms или более поздней WPF<sup>5</sup>. Причем разработка может вестись на различных языках программирования, таких как: C++, C#, Visual Basic.NET и F#. В данном программном пакете существует возможность интеграции с серверами баз данных Microsoft SQL Server и создания Silverlight web-приложений, также предусмотрен механизм работы с облачной платформой Windows Azure<sup>6</sup>. Для данного проекта был выбран язык C# - основной язык разработки для платформы Windows Phone.

Возможности среды могут быть значительно расширены за счет дополнительных модулей. Для разработки приложений, ориентированных на платформу Windows Phone, необходимо расширение Windows Phone SDK, которая включает в себя, помимо прочего, эмулятор мобильного устройства и примитивы, необходимые для разработки, а также позволяет протестировать программное обеспечение на реальном устройстве.

Создание приложений под операционную систему Windows Phone, подразумевает наличие соответствующей версии .NET Framework, установленной на машине разработчика. В данном случае используется версия 4.0, хотя могут быть использованы и более поздние редакции, поскольку различные версии программной платформы .NET обратно совместимы между собой, то есть приложение, разработанное для конкретной версии платформы, будет запускаться и в более поздних версиях этой платформы

---

<sup>5</sup>Windows Presentation Foundation

<sup>6</sup>Облачная платформа от компании Microsoft, реализующая модель Platform as a Service (PaaS)

В качестве шаблона проектирования архитектуры применяется шаблон MVVM (Model- View - ViewModel). Для облегчения построения клиентского приложения, согласно выбранному шаблону, используется MVVMLightToolkit – свободно распространяемый набор инструментов. Данный набор инструментов уже содержит примитивы, классы, события и команды, необходимые для начала разработки.

В качестве инструмента разработки графического интерфейса используется MicrosoftExpressionBlend. Приложение представляет собой интерактивный WYSIWYG<sup>7</sup>-редактор для дизайна интерфейсов, основанных на XAML<sup>8</sup>-приложениях, для WindowsPresentationFoundation и приложения Silverlight и входит в состав пакета графических программ Microsoft ExpressionStudio. ExpressionBlend поддерживает WPF, в котором используется метод OpenType и ClearType, основанный на двумерных и трехмерных элементах управления с аппаратным ускорением посредством Direct3D<sup>9</sup>.

---

<sup>7</sup>WhatYouSeeIsWhatYouGet - свойство прикладных программ или веб-интерфейсов, в которых содержание отображается в процессе редактирования и выглядит максимально близко похожим на конечный результат.

<sup>8</sup>eXtensible Application Markup Language — расширяемый язык разметки [для] приложений.

<sup>9</sup>API, разработанных для решения задач, связанных с визуализацией под ОС Microsoft Windows.



## 6.2 Декомпозиция задачи

Для создания сервиса необходимо реализовать два ключевых модуля: сервер-агрегатор данных и клиент, ориентированный на платформу WindowsPhone. Также нужно организовать передачу данных между сервером и клиентом и сервером и источниками данных. Таким образом, общая архитектура выглядит следующим образом:

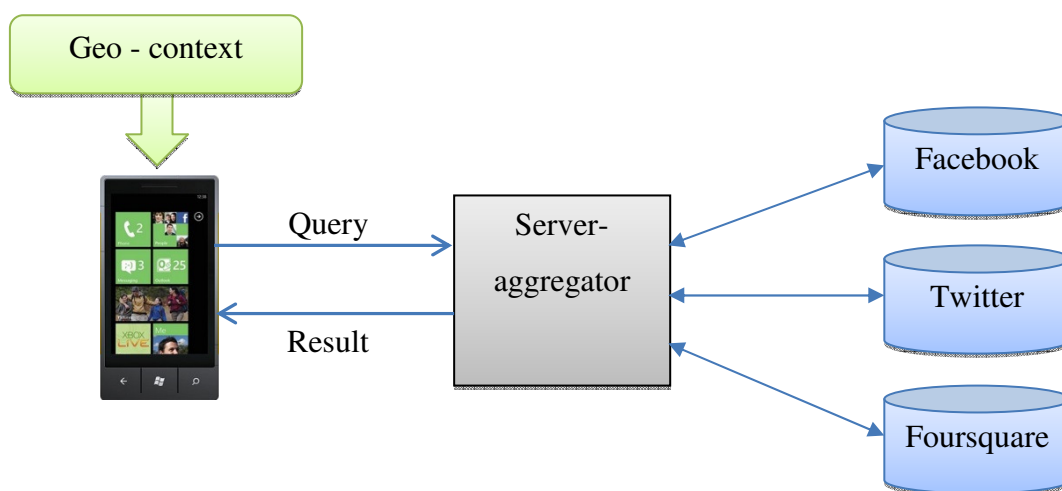


Рисунок 2: Общая архитектура сервиса

Такая трехступенчатая архитектура (клиент → сервер-агрегатор → источники данных) была выбрана по ряду причин:

- Она позволяет гибко интегрировать разнородные ресурсы в рамках единого логического звена.
- Выделение промежуточного этапа обработки позволяет уменьшить объем расчётов проводимых на клиенте.
- Также появляется возможность хранения и обработки больших объемов данных, что недоступно для мобильного устройства, ввиду ограничений аппаратного обеспечения.

### 6.3 Сетевой компонент

Сервер является сетевым компонентом сервиса. В рамках данной реализации прототипа сервиса именно сервер реализует два основных архитектурных слоя: слой предметных посредников и слой унификации данных. При этом важно отметить, что в общем случае все слои могут быть разделены и реализованы в виде отдельных программных модулей, при условии обеспечения взаимосвязи между ними.

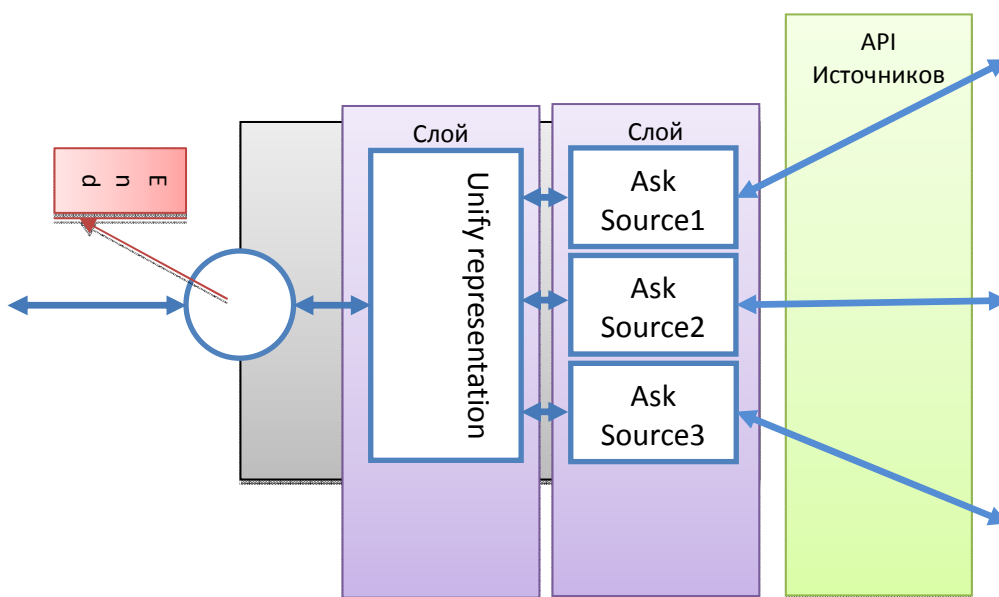


Рисунок 3: Реализация сетевого компонента

В пределах данной работы решено было остановиться на следующих источниках данных:

- Foursquare
- Twitter
- Facebook

Эти сети являются одними из наиболее крупных социальных сетей. Каждая из этих сетей имеет возможность отметки пользователем своего географического местоположения, и предоставляет публичный, открытый API для доступа к хранящейся информации. Более того в случае с Foursquare– еще и присутствует гео-ориентированная направленность, которая является основополагающей.

Важно заметить, что предоставляемый каждым сервисом API индивидуален. Создание единого модуля для всех источников – универсального посредника – приведет к большим проблемам в расширяемости. К тому же API довольно часто меняется: вводятся новые возможности, а старые модифицируются или удаляются. В случае одного глобального посредника изменение в работе с одним источником может привести к неработоспособности всей системы.

Именно по этой причине выделяются три различных посредника, которые работают только со своим источником. На Рисунок 3 посредники обозначены как элементы AskSource 1..3.

С точки зрения программного кода, каждый из посредников может быть классом или же набором классов. В данном проекте каждый посредник – это один класс. Каждый подобный класс имеет методы для запроса данных у источника. Опять же стоит заметить, что каждый класс скрывает сложность доступа к тому или иному ресурсу. Результатом работы является документ в формате JSON, содержащий найденную информацию.

В дальнейшем этот документ попадает на обработку к классу унификатору. В этом классе определены обозначенные ранее теоретико-множественные операции (объединение, пересечение, разность). Получив данные от слоя посредников, происходит вычисление пересечения документов, для уточнения этого процесса важно изучить формат ответов всех источников и определить словарь синонимов, в противном случае возможно формирование результатов из элементов с одинаковым смыслом, но разным названием.

## 6.4 Клиентский компонент

Клиентский компонент реализует контекстный слой. Основной задачей этого модуля является сбор контекстных данных и отправка запросов к сетевому компоненту. Клиент представляет собой приложение, ориентированное на платформу WindowsPhone. Ниже приведена схема клиентского приложения.

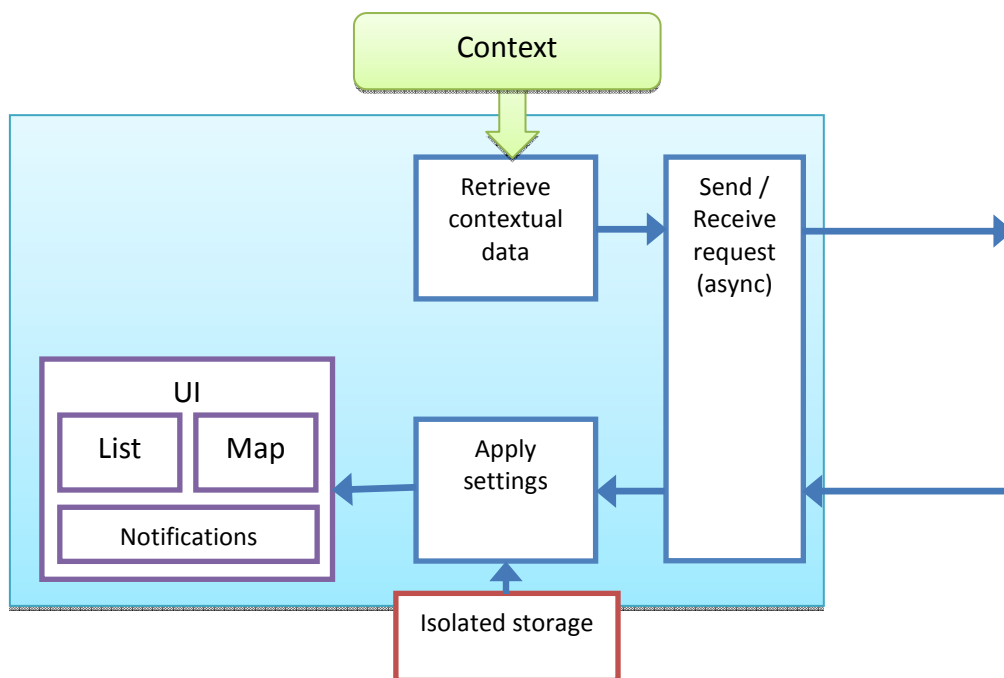


Рисунок 4: Реализация клиентского компонента

Все современные устройства с этой операционной системой имеют возможность определения собственного географического местоположения. Клиентское приложение позволяет не только определять местоположение, когда оно активно, но и в фоновом режиме, помимо этого оно позволяет хранить и применять пользовательские настройки. Программное обеспечение также позволяет осуществлять запросы к сетевому компоненту асинхронно.

В качестве программной платформы для разработки этого компонента выступает Microsoft Silverlight. Ключом к возможностям Silverlight, как и ко всей технологии представления WPF платформы Microsoft .NET Framework 3.0 (и выше), является XAML.

Одним из основных шаблонов проектирования для данной платформы является MVVM.

### 6.4.1 Шаблон проектирования

Клиентская часть создана с применением шаблона проектирования MVVM. Данная модель подразумевает разделение архитектуры приложения на три уровня:

1. View: уровень представления. К этому уровню можно отнести графический интерфейс.
2. Model: уровень модели (реализация бизнес-логики). Модель представляет собой фундаментальные данные, необходимые для работы приложения;
3. ViewModel: связующий уровень;

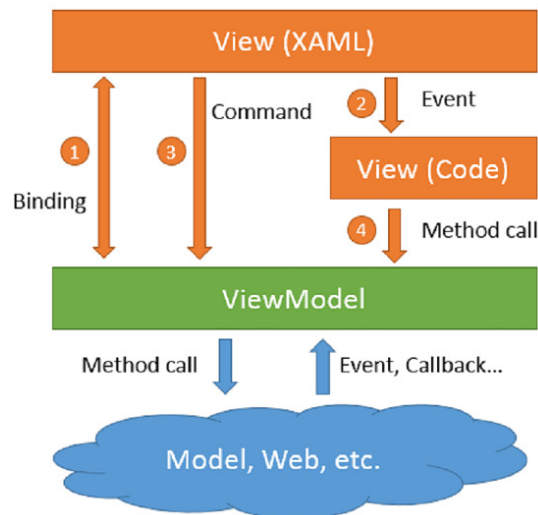


Рисунок 5: Слои MVVM и зависимости между ними

Самый важный момент Silverlight/WPF, делающий MVVM очень удобным шаблоном, – это инфраструктура привязки данных. За счет привязки свойств представления к модели представления (ViewModel) получается слабое связывание этих компонентов, что полностью освобождает разработчика от необходимости писать в модели представления код, непосредственно обновляющий представление. Система привязки данных поддерживает также проверку допустимости ввода, обеспечивающую стандартный путь передачи ошибок проверки допустимости представлению.

В случае данного программного обеспечения, отображение списка интересующих людей и демонстрация карты относятся к уровню представления (View), запросы к системе и прием результатов к уровню модели (Model).

В результате применения этого шаблона достигается гибкость разработки программных модулей клиентского программного обеспечения. Разработка графического интерфейса может быть всецело отделена от логики и привязана к ней в последний момент. В тоже время при разработке уровня модели не нужно думать о ее представлении в графическом виде, функцию привязки возьмет на себя уровень модели представления.

### 6.4.2 Фоновое обновление и нотификации

Клиентское программное обеспечение способно отслеживать положения даже будучи в фоновом режиме. Для реализации подобного функционала в операционной системе WindowsPhone предусмотрено несколько механизмов, но для нужд данного приложения требуется только один из них:

- `Scheduledtaskbackgroundagent`. Данный агент позволяет выполнять определённый код в фоновом режиме, даже если приложение, породившее данный агент, не запущено. Выделяют два типа заданий, которые могут быть выполнены таким агентом:
  - Периодические задания (`periodictask`), которые выполняются регулярно, но в короткий промежуток времени.
  - Ресурсо-интенсивные (`resource-intensive`), могут выполняться довольно долго, при условии, что выполнение задания не будет приводить к сильному замедлению работы устройства.

К ограничениям системы относится тот факт, что одновременно программа может запустить только один агент. Поэтому, несмотря на то, что существует еще один тип агентов - `Schedulednotificationbackgroundagent`, позволяющий запускать уведомления по расписанию, нотификацией пользователя и обновлением информации должен заниматься один и тот же агент.

Поскольку обновление информации о местоположении не является ресурсо-интенсивной задачей (при условии включенного приемника спутникового сигнала), а запросы к сетевому компоненту системы посылаются асинхронно, был выбран `Scheduledtaskbackgroundagent`, выполняющий периодические задания.

Существует ряд ограничений натакого рода агенты:

- Периодические задания могут запускаться на исполнение не чаще одного раза в 30 минут.
- Время выполнения ограничено 25 секундами.
- Объем используемой памяти не должен превышать 11 мегабайт.



- Каждые две недели агент должен быть запущен повторно, иначе его выполнение будет прекращено.

Все указанные ограничения были учтены при разработке системы. Сетевой компонент выполняет проверку объема передаваемых данных, также выполняет обновление списка местоположения интересующих людей, не дожидаясь запроса пользователя, таким образом, происходит кеширование данных.

Касательно нотификаций, данный программный компонент использует два типа нотификаций. Первый из них -это LiveTiles или живые плитки – основополагающий элемент пользовательского интерфейса операционной системы. Второй – Toastmessages (всплывающие уведомления).

Каждый из них несет в себе свою функциональность. Если живая плитка позволяет показывать количество людей поблизости и вывести имена двух ближайших, то всплывающее уведомление лишь показывает, сколько человек находится неподалеку.

## 6.5 Иллюстрация результатов работы

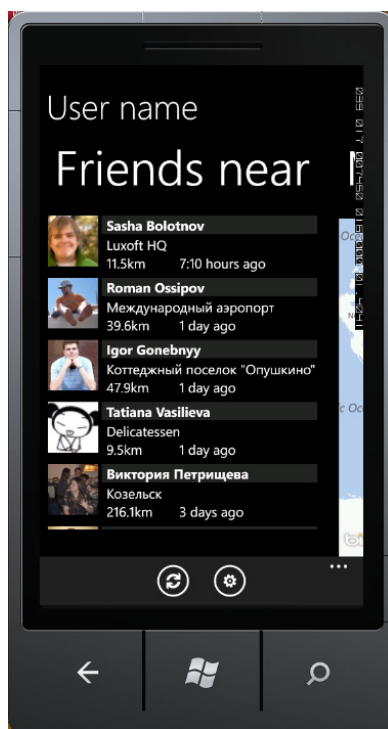


Рисунок 6: Демонстрация списка людей с указанием расстояния до них

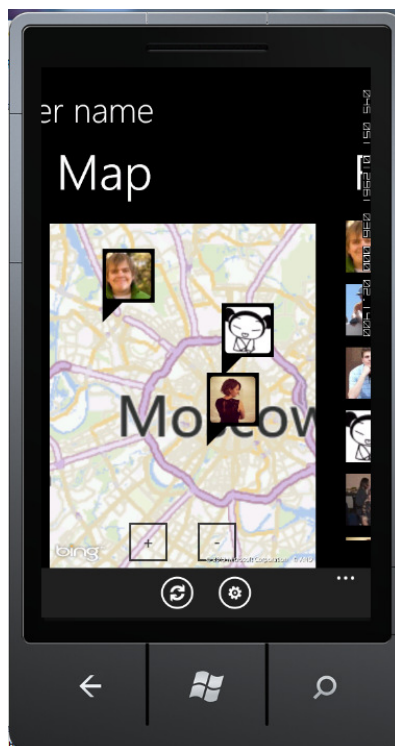


Рисунок 7: Демонстрация интересующих людей на карте

## 6.6 Возможные направления развития

Данный вариант реализации сервиса не является единственным. Целью была лишь демонстрация работоспособности предложенной архитектуры. Функциональность каждого из компонентов может быть значительно расширена.

В перспективе получаемые из источников данные можно сохранять в некоторой базе данных и проводить статистические исследования. Применяя различные методы исследования результатов, можно будет предсказывать появление тех или иных людей в различных местах.

Например, зная, что человек довольно часто появляется в одних и тех же местах на протяжении дня, можно выдвинуть предположения о месте его работы и проживания, определить любимые заведения.

Проблему масштабируемости можно решить путем использования облачных технологий. Таким образом затребовать под нужды сетевой части ровно столько ресурсов, сколько это необходимо.

## **7 Заключение**

В рамках данной работы был приведен вариант универсальной архитектуры для решения задач поиска и агрегации гео-контекстных данных из различных источников. Выделены основные элементы архитектуры сервиса и указана их функциональность и способы обмена данными между ними.

Разработано отображение приведенной архитектуры на элементы платформы .NET на которую опирается операционная система WindowsPhone. Созданы модули, активно использующие возможности платформы, такие как фоновые агенты, платформа для визуализации Silverlight, а также технологии межмодульного взаимодействия (WCF).

Также был создан действующий прототип сервиса, опирающийся на данную архитектуру и реализующий необходимый функционал. Создана клиентская и серверная составляющая с широким привлечением возможностей платформы .NET. Реализовано извлечение гео-контекстных данных с использованием стандартных средств, прилагаемых операционной системой, а также фоновая работа клиентского модуля за счет активного привлечения фоновых агентов.

## 8 Библиография

- [1] B. Schilit, M. Theimer Disseminating active map information to mobile hostsnetwork. // IEEE, vol. 8, no. 5, 1994.
- [2] K. Dey Understanding and using context. // Personal Ubiquitous Computing, vol. 5, January 2001.
- [3] J. Giles, Context-Aware and Location Systems, University of Cambridge, January 1998.
- [4] C. Schmidt, Context-Aware Computing, Berlin Institute of Technology, 2011.
- [5] Nokia Research Center, Nokia Technology Insights series, January 2009.
- [6] Chin, Finding Cohesive Subgroups and Relevant Membership the Nokia Friend View Mobile Social Network, Nokia Research Center, 2009.
- [7] Biancalana, F. Gasparetti, A. Micarelli, G. Sansonetti, An Approach to Social Recommendation for Context-Aware Mobile Services, Roma Tre University, 2011.
- [8] Mediation. URL: <http://synthesis.ipi.ac.ru/synthesis/student/oodb/mediation.ppt>
- [9] Виды предметных посредников и методология интеграции неоднородных ресурсов в посредниках. URL: <http://synthesis.ipi.ac.ru/synthesis/student/SpecSem/1-2>
- [10] Регистрация ресурсов в посреднике. URL: <http://synthesis.ipi.ac.ru/synthesis/student/SpecSem/3-2>
- [11] Бильчук А.А., Намиот Д.Е. Geo Messages для Java телефонов // INJOIT, Vol. 1, No. 1, 2013, с. 16-20
- [12] Громаков Ю.А., Северин А.В., Шевцов В.А. Технологии определения местоположения в GSM и UMTS. М.: Эко-Трендз, 2005.
- [13] В. А. Ильин, В. А. Садовничий, Бл. Х. Сендов. Математический анализ М.: Проспект, 2006.
- [14] Смит Д. Приложения WPF с шаблоном проектирования модель-представление-модель представления. //MSDN Magazine, Февраль 2009
- [15] Морони Л. Знакомство с Silverlight. // Корпорация Майкрософт, 30 апреля 2007
- [16] L. Bugnion, Messenger and View Services in MVVM. // MSDN Magazine, March 2013

- [17] Multitasking for Windows Phone. URL:[http://msdn.microsoft.com/ru-ru/library/windowsphone/develop/hh202866\(v=vs.105\).aspx](http://msdn.microsoft.com/ru-ru/library/windowsphone/develop/hh202866(v=vs.105).aspx)
- [18] Background agents for Windows Phone.  
URL:[http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh202942\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh202942(v=vs.105).aspx)
- [19] Bing Maps API. URL:<http://www.bingmapsportal.com/isdk/ajaxv7>
- [20] Fousquare API. URL:<https://developer.foursquare.com/docs/>
- [21] Twitter API. URL:<https://dev.twitter.com/>
- [22] Facebook API. URL:<http://developers.facebook.com/>
- [23] Юсуф-заде Г., Намиот Д. Автоматический гео-контекстный анализ на мобильных платформах //International Journal of Open Information Technologies. – 2013. – Т. 1. – №. 1. – С. 10-15.
- [24] D. Namiot, “Geo messages”, Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2010 International Congress pp. 14-19  
DOI: 10.1109/ICUMT.2010.5676665