

ОПЫТ РЕАЛИЗАЦИИ КАРТОГРАФИЧЕСКИХ ВЕБ-СЕРВИСОВ ДЛЯ ЗАДАЧ МОНИТОРИНГА ТРАНСПОРТНЫХ СРЕДСТВ НА ОСНОВЕ СВОБОДНО РАСПРОСТРАНЯЕМОГО ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ

А.В. Токарев

Введение

Системы мониторинга на основе технологий спутникового позиционирования GPS/ГЛОНАСС обеспечивают эффективный контроль за движением транспорта и управление транспортным комплексом города или автопарком предприятия. Одним из существенных модулей системы мониторинга является блок работы с картографической информацией. Он обеспечивает не только наглядное отображение местоположений транспортных средств, но и решение ряда вспомогательных задач связанных с адресным поиском и дорожной сетью. Применение сервис-ориентированной архитектуры позволяет выделить и разнести отдельные компоненты программного обеспечения. Такое решение повышает масштабируемость системы и позволяет развивать отдельные модули независимо друг от друга.

В настоящей работе рассматривается реализация нескольких гео-сервисов – сервиса геокодирования и сервиса поиска маршрута. Геокодирование (geocoding) – процесс преобразования адресов пространственных объектов в их географические координаты. Обратный процесс, преобразование точки на карте в читаемый для пользователя адрес, известен как обратное геокодирование. Функции геокодирования позволяют "привязывать" базы данных, которые ведет большинство ведомств, обслуживающих урбанизированные территории и население, к картам территорий [1].

Популярные картографические порталы, такие как Google Maps, Яндекс.Карты, Yahoo! Maps, и ряд других предоставляют API, с помощью которого можно добавлять картографические возможности на сторонние сайты. В том числе и возможности геокодирования. Обычно поддерживается несколько вариантов программных интерфейсов – прямые запросы по HTTP протоколу (GET – XML), а также JavaScript библиотеки, которые облегчают использование веб-сервиса.

Open Geospatial Consortium разрабатывает стандарты в области сервисов позиционирования - Open Location Services (OpenLS). В настоящее время созданы спецификации сервисов геокодирования (Geocoder Service), каталога товаров и услуг (Directory Service), сервиса поиска маршрутов (Route Service), и ряда других.

Заслуживает внимания проект OpenStreetMap (OSM) (<http://www.openstreetmap.org>). Это свободный проект по совместному развитию общедоступных карт и схем городов, улиц, дорог с помощью ручного или автоматического ввода данных, а также получения данных с портативных GPS-приёмников. Помимо уличной сети в OSM содержится и отображается на карте множество других данных. На основе идей и данных OpenStreetMap развивается большое количество смежных проектов.

Несмотря на существующие стандарты и решения, есть потребность в создании картографических сервисов, это связано с несколькими факторами:

- лицензионные ограничения публичных картографических сервисов;
- недостаточная полнота исходных пространственных данных;
- невозможность применения собственных алгоритмов
- и др.

Алгоритм геокодирования

Существуют различные методы прямого и обратного геокодирования [2]. В значительной степени применяемые алгоритмы зависят от набора и полноты исходных данных, на основе которых выполняется геокодирование. В рамках этой работы предполагается, что доступен следующий набор данных:

- объекты с адресной привязкой (районы, города и населенные пункты, улицы, здания);
- дорожная сеть (узлы дорожной сети, сегменты дорог).

На текущем этапе были использованы достаточно простые методы геокодирования. Для определения координат объектов по адресу решено использовать геокодирование по объектам. Искомый адрес сопоставляется с адресами существующих объектов, и в качестве результата возвращаются координаты геоцентра наиболее подходящего объекта.

Считаем, что входной запрос содержит адрес объекта, записанный в свободной форме. Предлагается следующий алгоритм прямого геокодирования, на основе полнотекстового поиска [3]:

1. Разбиваем запрос на отдельные лексемы
2. Формируем полнотекстовый запрос из лексем по логике "И"
3. Выполняем полнотекстовый поиск в таблице всех объектов по полю с полным адресом, накладываем дополнительные ограничения на район, город, улицу.

4. Сортируем результат по классу объекта (от крупных образований к более мелким) и релевантности, нормированной на длину адреса.

5. Выбираем первые N объектов, как наиболее подходящих, и выдаем информацию о них пользователю.

Для определения адреса по координатам предлагается смешанный подход. В границах населенных пунктов ориентируемся по ближайшим зданиям, а за пределами населенных пунктов – ориентируемся по дорожной сети (рис. 1). Во втором случае адресом точки могут являться разные сущности:

- адрес ближайшего населенного пункта + расстояние до него по дорожной сети;
- адрес ближайшего крупного населенного пункта + расстояние до него по дорожной сети;
- наименование ближайшей федеральной трассы + километровый отсчет вдоль нее + расстояние до трассы.

На текущем этапе было решено использовать первые два варианта.

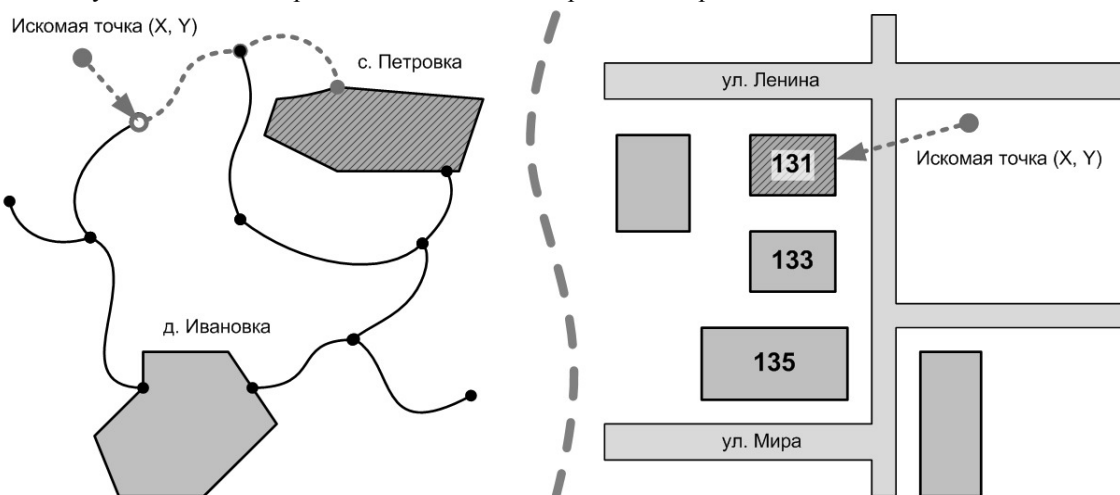


Рис. 1 Схема обратного геокодирования

Алгоритм обратного геокодирования:

1. Разбираем запрос, определяем координаты искомой точки.
2. Определяем, попадает исходная точка в населенный пункт (переход на шаг 3) или нет (переход на шаг 4).
3. Точка находится в населенном пункте – ориентируемся по зданиям.
 - 3.1. Выполняем поиск ближайших зданий в населенном пункте.
 - 3.2. Переход на шаг 5
4. Точка находится за границами населенного пункта – ориентируемся по дорожной сети.
 - 4.1. Определяем ближайший сегмент дороги
 - 4.2. Получаем 2 узла сегмента, а также заранее вычисленные "адреса" (населенный пункт + расстояние до него по дорожной сети) этих узлов.
 - 4.3. Вычисляем кратчайший путь по прямой к ближайшему сегменту и точку съезда с дороги.
 - 4.4. Вычисляем все адреса точки съезда по адресам узлов сегмента. Для этого добавляем расстояние от точки съезда до узла к расстояниям в адресах узлов. Полученный список группируем по населенным пунктам и выбираем только адреса с минимальным расстоянием.
5. Выдаем информацию о найденных адресах пользователю.

Алгоритм поиска маршрута

Поиск маршрута на дорожной сети выполняется по двум указанным точкам – начальной и конечной. Предполагаем, что они могут быть заданы как географическими координатами, так и адресом в текстовом виде. В последнем случае используется подсистема геокодирования для определения их координат.

На следующем шаге необходимо выполнить переход от географических координат к сегментам и узлам дорожного графа. В городах, где дорожные сегменты короткие и узловые точки расположены близко друг к другу, достаточно хорошо работает простейший алгоритм – выбирается ближайший узел графа к исходной точке. Однако вне населенных пунктов узлы дорожной сети могут располагать далеко друг от друга, и подобный алгоритм будет давать существенную погрешность. Поэтому используется более точный метод, основанный на поиске ближайшего сегмента дороги и вычислении точек "съезда/заезда" на дорожную сеть.

После определения начального и конечного узла дорожного графа, применяются уже алгоритмы на графах. В задаче о кратчайшем пути (shortest-paths problem) задается взвешенный ориентированный граф $G = (V, E)$ с весовой функцией $w: E \rightarrow R$, отображающей ребра на их веса, значения которых выражаются действительными числами. Вес пути равен суммарному весу входящих в него ребер. Вес каждого из ребер можно интерпретировать не только как расстояние, но и как другую метрику [4]. Задавая разные функции веса,

мы получаем оптимальные маршруты по разным критериям оптимальности. Если в качестве весов задавать расстояние вдоль сегмента дороги – получим кратчайший маршрут, если условное время передвижения – наискорейший. Аналогично можно учитывать и другие особенности маршрута.

Известно большое количество алгоритмов поиска, позволяющих учитывать различные условия задачи, в том числе и методы с предварительной обработкой (ALT, REACH) [5], предназначенные для поиска на графах с большим количеством узлов.

В данной работе опробовался алгоритм поиска оптимального пути на взвешенном графе между двумя узлами A*. A* – это алгоритм поиска в ширину, но просматриваются сначала те маршруты, которые "кажутся" ведущими к цели. Для этого используется эвристическая функция расстояния (использовалась евклидова метрика).

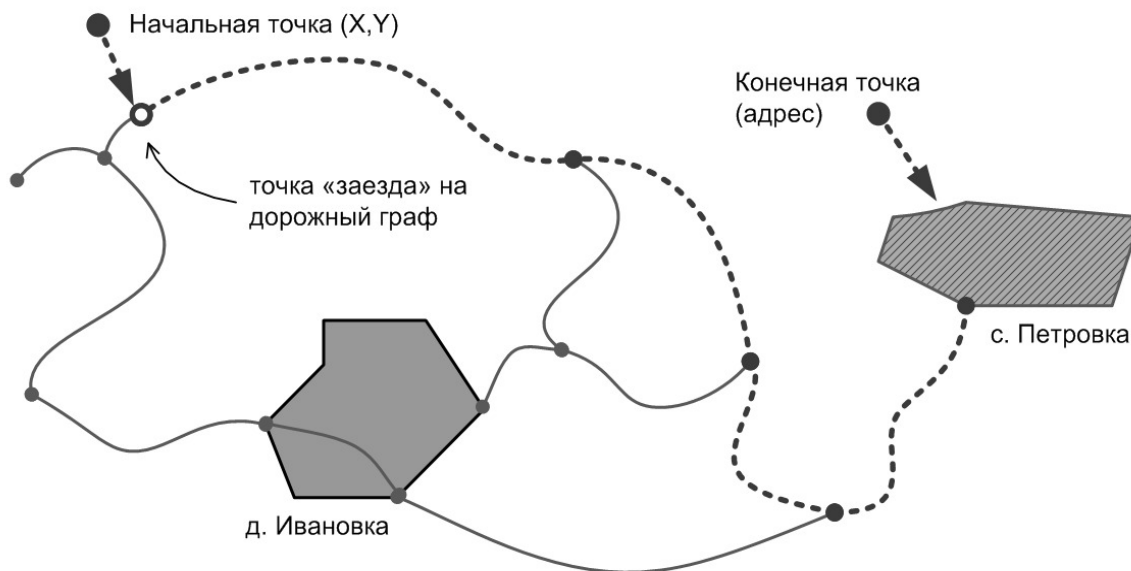


Рис. 2 Схема поиска маршрута

Алгоритм поиска маршрута (рис. 2):

1. Разбираем запрос, определяем начальную и конечную точки.
2. Если точка (начальная/конечная) задана координатами переход на шаг 3
- 2.1. Выполняем процедуру геокодирования и получаем координаты
3. Для начальной и конечной точек определяем ближайшие сегменты дорог, ближайшие узлы и точки съезда/заезда.
4. Выполняем поиск кратчайшего маршрута на графе между начальным и конечным узлами
5. Добавляем к найденному маршруту сегменты дорог заезда и съезда
6. Выдаем информацию о найденных адресах пользователю.

База данных

Для хранения реперных объектов с адресной привязкой, а также дорожной сети разработана концептуальная модель базы данных (рис. 3). На ее основе сформирована физическая модель и создана база данных. Основные сущности:

- Объект (район, населенный пункт, здание). У объекта хранится полный адрес, геометрия и служебные поля. По геометрии построен пространственный индекс, что обеспечивает достаточно быстрый поиск.
- Узлы и сегменты дорожного графа. Сегмент дороги хранит ссылки на начальный и конечный узлы, геометрию, а также стоимость проезда в прямом и обратном направлениях.
- Ближайшие населенные пункты. Для каждого из узлов дорожного графа были вычислены несколько "адресов" – расстояние до ближайшего населенного пункта и расстояние до ближайшего крупного города. Для этого был применен модифицированный алгоритм Дейкстры для поиска кратчайшего пути на графе.

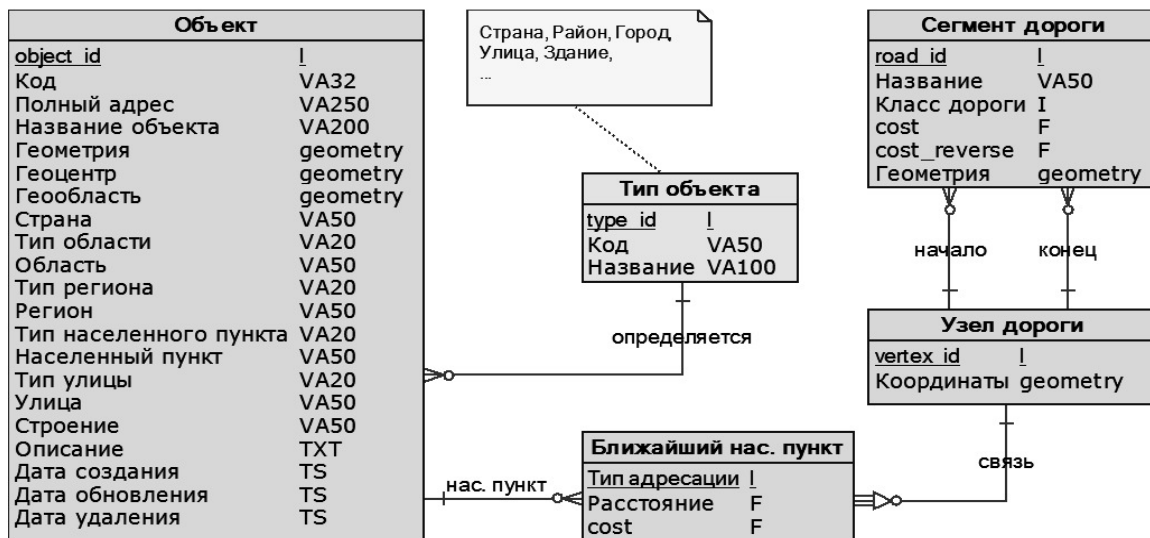


Рис. 3 Концептуальная модель базы данных

Технологии

Используются различные способы построения веб-сервисов. На сегодняшний день наибольшее распространение получили следующие протоколы и подходы в реализации веб-сервисов: XML-RPC, SOAP, REST [6].

Самые первые веб-сервисы часто использовали в качестве языка запросов простые GET/POST запросы по HTTP протоколу, аналогичные запросам веб-браузера, а в качестве ответов обычно использовались XML документы. В дальнейшем этот подход сформировался в виде архитектурного стиля построения веб-сервисов REST (Representational State Transfer). Это набор общих принципов построения веб-сервисов с определенными приоритетами: масштабируемость, независимость от платформы, расширяемость. Сущность подхода в том, что в системе выделяются базовые ресурсы, адресуемые по уникальному идентификатору URL, к которым применимы запросы HTTP протокола (GET, POST, DELETE, и др.). Самой известной системой, построенной в значительной степени по архитектуре REST, является современная Всемирная паутина (WWW).

Необходимость стандартизировать способ представления запросов и ответов для веб-сервисов привела к созданию стандарта SOAP (Simple Object Access Protocol), а так же ряда сопутствующих стандартов – языка описания веб-сервисов WSDL, распределенной системы регистрации веб-служб UDDI, и др. SOAP изначально разрабатывался как замена протоколу XML-RPC, но позже был значительно доработан, и стал протоколом-стандартом для построения веб-сервисов уровня предприятия.

Как правило, веб-сервисы, которые в основном предназначены для поиска и получения информации, лучше приспособлен REST подход. Поэтому мы будем ориентироваться на него.

Для хранения данных выбрана СУБД PostgreSQL (<http://www.postgresql.org>) – свободно распространяемая объектно-реляционная система управления базами данных, наиболее развитая из открытых СУБД в мире и являющаяся реальной альтернативой коммерческим базам данных. Немаловажным преимуществом является наличие дополнительных модулей, которые облегчают решение задач геокодирования:

- PostGIS (<http://www.postgis.org>) – это свободная ГИС библиотека, которая позволяет работать с географическими объектами и функциями в базе данных PostgreSQL. PostGIS соответствует OpenGIS стандартам, разработанным Открытым Географическим Сообществом (OGC). Поддерживается более 300 различных функций для работы с векторными данными, пространственные индексы, популярные обменные форматы и многое другое.
- pgRouting (<http://pgrouting.postlbs.org>) – модуль, который расширяет функции базы данных PostgreSQL/PostGIS, добавляя в них функции планирование пути. pgRouting позволяет создавать топологии и решать различные задачи по планированию маршрута (например, расчет кратчайшего пути).

Подготовка и импорт данных

Исходными картографическими данными, являлись слои, подготовленные в ООО "ТОРИНС" (www.torins.ru):

- районы Красноярского края и республики Хакасии;
- населенные пункты (полигональные);
- дорожная сеть между населенными пунктами и уличная сеть в крупных городах;
- дома с почтовыми адресами.

С помощью утилиты shp2pgsql.exe из комплекта PostGIS исходные слои были преобразованы из формата ESRI Shape в SQL-сценарии и созданы временные таблицы базы данных. Эти таблицы использовались для промежуточных преобразований и фильтрации данных.

Полигональные объекты из слоев районы, населенные пункты, дома были добавлены в таблицу объектов. Для каждого объекта сгенерирован полный адрес.

Для решения задачи обратного геокодирования по предложенному алгоритму за границами населенных пунктов необходимо иметь связную и топологически корректную дорожную сеть. Исходные данные имели не совсем корректную топологию – были примыкающие дороги без образования узла, а также отсутствовали узлы дороги в местах пересечения с границами населенных пунктов. С помощью функций PostGIS были выполнены разбиения объектов слоя дорог на сегменты, выделены узлы, а также проставлены связи.

Общий объем данных: полигональных объектов с адресной привязкой – 17 тыс. объектов, 32 тыс. сегментов дорог и 26 тыс. дорожных узлов.

Программный интерфейс

Веб-сервис геокодирования был реализован на основе REST подхода. Запрос передается в виде GET параметров, результат формируется в виде XML. Базовая структура запроса следующая: `http://<сервер>/<сервис>?арі=<арі>&srіd=<проекция>&<дополнительные параметры>`, где:

- сервис – код сервиса (geocode – сервис геокодирования, route – сервис поиска маршрутов);
- arі – версия интерфейса, параметр предназначен для поддержки устаревших версий программного интерфейса;
- srіd – код проекции (EPSG код);
- output – формат выдаваемого ответа (xml - XML (по-умолчанию), json - в разработке).

Если <сервис>=geocode, то в зависимости от значения параметра "запрос" выполняется прямое или обратное геокодирование. Поддерживаются следующие параметры:

- q – запрос в виде координат или текстового адреса;
- limit – ограничение на количество выдаваемых результатов;
- aom – степень детализации выдаваемых результатов в виде битовой маски:
 - 1 - информация о ближайшем объекте /object
 - 2 - информация об адресе объекта object/address/text
 - 4 - детализация адреса объекта object/address/*
 - 8 - размещение объекта /object/extent
 - 16 - смещение исходной точки от объекта /offset
 - 128 - адрес объекта со смещением в виде одной строки /description
- region – название района;
- city – название населенного пункта;
- street – название улицы.

Если параметр "запрос" задан во множественном виде (q[]=<запрос1>&q[]=<запрос2>&q[]=...), то выполняется пакетное геокодирование всех значений. Структура выдаваемых результатов показана на рис. 4.

Если <сервис>=route, используется модуль поиска маршрутов. Параметры модуля:

- from – описание точки старта, заданная координатами или адресом;
- to – описание конечной точки;
- limit – ограничение на количество выдаваемых результатов;
- aom – степень детализации выдаваемых результатов в виде битовой маски:
 - 1 - информация о ближайшем объекте /object
 - 2 - информация об адресе объекта object/address/text
 - 4 - детализация адреса объекта object/address/*
 - 8 - размещение объекта /object/extent
 - 16 - смещение исходной точки от объекта /offset
 - 128 - адрес объекта со смещением в виде одной строки /description
- rom – режим выдачи информации о маршруте в виде битовой маски:
 - 1 - выдавать геометрию
 - 2 - выдавать инструкции

```

<result>
  <object id="0c358595-2313-4dc2-a979-99d1b749dfda">
    <type code="building">Здание</type>
    <address>
      <text>Россия, Красноярский край, г Красноярск, ул. Энергетиков, 58</text>
      <country>Россия</country>
      <area type="">Красноярский край</area>
      <region type=""/>
      <place type="r" okato="">Красноярск</place>
      <street type="">Энергетиков</street>
      <building>58</building>
    </address>
    <extent>BOX(92.999160 56.001037,93.000808 56.002609)</extent>
  </object>
  <offset>
    <distance type="straight" meters="100">0.1 км</distance>
    <azimuth degrees="182.2">запад</azimuth>
    <description>0.1 км на запад</description>
  </offset>
  <description>0.1 км на запад от ул. Энергетиков, 58, г Красноярск</description>
</result>

```

Рис. 4 Фрагмент ответа сервера на запрос геокодирования

Заключение

Разработанное решение проходит тестовую эксплуатацию в нескольких проектах. В настоящее время веб-сервис активно используется в системе спутникового мониторинга транспорта "Регнасс" (<http://www.torins.ru/regnass/>). Система предназначена для оперативного мониторинга транспортных средств, используемых для перевозки школьников, машин "Скорой помощи" и автобусного парка пригородных перевозок в Красноярском крае. Сервис обеспечивает оперативное определение адреса транспортных средств по их координатам, построение маршрутов и определение расстояний по дорожной сети между двумя точками.

Можно выделить следующие преимущества созданного веб-сервиса:

- возможность использования "своих", более полных пространственных данных, по сравнению с публичными источниками;
- поддержка преобразований координат в проекцию, заданную клиентом;
- настраиваемая детализация выдаваемых данных;
- поддержка пакетного геокодирования нескольких адресов за один запрос.

Рассматриваются несколько направлений развития сервиса: доработка алгоритмов геокодирования и поиска маршрутов, переход на стандартные форматы данных OpenLS, добавление механизма разграничения доступа, увеличение скорости работы сервиса. Кроме этого обсуждается развитие предложенных решений в область интеллектуальных транспортных систем.

ЛИТЕРАТУРА:

1. Капралов Е.Г., Кошкарев А.В., Тикунов В.С. и др. Геоинформатика: Учеб. для студ. вузов / Под ред. Тикунова В.С. – М.: Издательский центр "Академия", 2005. – 480 с.
2. Goldberg, D.W. A Geocoding Best Practices Guide. Springfield, IL: North American Association of Central Cancer Registries, 2008. 261 pages.
3. Токарев А. В. Построение веб-сервиса геокодирования // Тезисы докладов XI Всероссийской конференции молодых ученых по математическому моделированию и информационным технологиям. – Красноярск, 2010. – С. 70-71.
4. Кормен, Т., Лейзерсон, Ч., Ривест, Р., Штайн, К. Алгоритмы: построение и анализ = Introduction to Algorithms / Под ред. И. В. Красикова. — 2-е изд. — М.: Вильямс, 2005. — 1296 с. — ISBN 5-8459-0857-4
5. Goldberg A. V., Kaplan H., Werneck R. Reach for A*: Efficient Point-to-Point Shortest Path Algorithm. SIAM Workshop on Algorithms Engineering and Experimentation (ALENEX '06), Miami, 2006.

6. Pautasso, C.; Zimmermann, O.; Leymann, F. RESTful Web Services vs. Big Web Services: Making the Right Architectural Decision // 17th International World Wide Web Conference (WWW2008). – Beijing, China, 2008. – P. 805-814.