

УДК 004.8

# ИНТЕЛЛЕКТУАЛЬНАЯ СИСТЕМА УПРАВЛЕНИЯ ЛОГИСТИЧЕСКОЙ СЕТЬЮ ПРИ СОВМЕСТНОМ ИСПОЛЬЗОВАНИИ АВТОТРАНСПОРТА

**А. М. Кашевник,**

канд. техн. наук, старший научный сотрудник

Санкт-Петербургский институт информатики и автоматизации РАН

Рассматривается применение концепции интеллектуальных пространств для разработки интеллектуальной системы управления логистической сетью при совместном использовании автотранспорта. Система базируется на онтологии интеллектуального пространства, построенной на основе интеграции соответствующих частей онтологий мобильных устройств участников системы. Предложен алгоритм для поиска попутчиков для водителей на основе совпадения путей водителей и начальных и конечных точек путей пассажиров. Ввиду достаточно большой размерности задачи для ее решения предлагается использование эвристик, позволяющих снизить ее размерность, что дает возможность существенно сократить время работы системы.

**Ключевые слова** — онтология, интеллектуальное пространство, транспортная логистика, карпул.

## Введение

Многие предприятия, находящиеся вне зоны покрытия сети общественного транспорта, в рамках предоставляемого социального пакета включают доставку сотрудников до места работы. Самыми яркими примерами таких предприятий являются автомобильные заводы, расположенные за границами населенных пунктов. Организация данного процесса имеет плюсы как для работника, так и для работодателя: работнику не нужно беспокоиться о ежедневном проезде до места работы и расходовать дополнительные средства, при этом он прибывает на работу вовремя, что положительно сказывается на производственном процессе. Подобная услуга может быть организована как силами самого предприятия, так и заключением договора со сторонней организацией, осуществляющей пассажирские перевозки. В любом случае, для предприятия это означает внесение в бюджет дополнительной статьи расходов, связанной либо с содержанием дополнительных единиц автотранспорта, либо с арендой транспорта у сторонней организации.

В организациях, не предоставляющих такую услугу, сотрудники добираются до места работы на общественном транспорте либо на личном автомобиле. Существующая сеть маршрутов общественного транспорта не всегда достаточно полно

покрывает населенный пункт, и пассажирам зачастую приходится длительное время добираться до остановок либо ощутимо долго ждать общественного транспорта. Стремясь обеспечить независимость от маршрутов и расписания общественного транспорта, все больше людей переходят на личный транспорт. Увеличение количества автомобилей ведет к перегруженности существующей сети дорог, росту аварийности и ухудшению экологической обстановки в регионе.

Альтернативой развитию сети общественного транспорта и услуги доставки сотрудников до места работы может являться совместное использование автотранспорта, или карпул [1] (carpool, также известный как ride-sharing, lift-sharing и covoiturage). Главное отличие карпула от традиционных способов пассажироперевозок состоит в том, что в качестве используемого транспорта выступают личные автомобили участников дорожного движения. Преимущества данной системы очевидны: на дорогах уменьшается число транспортных средств, благодаря чему уменьшается число заторов и вероятность аварий, существенно снижается выброс парниковых газов и уменьшается количество припаркованных автомобилей на улицах города. Для участников карпула становится очевидным выигрыш в стоимости и удобстве поездки: используется только один автомобиль, следовательно, снижаются об-

щие расходы на топливо, ремонт и обслуживание. К тому же возникает возможность дополнительного общения людей.

В данной работе предлагается интеллектуальная система управления логистической сетью, объединяющей совместно используемые транспортные средства, главной задачей которой является поиск попутчиков для водителей на заданном маршруте. Система основана на использовании интеллектуальных пространств [2]. Предполагается, что каждый пассажир и водитель имеют мобильное устройство, которое с соответствующим программным обеспечением становится частью этого пространства.

### Существующие системы карпула

Идеи карпула существуют с 1970 г., однако повсеместное использование и рост их популярности стали возможны только с развитием сети Интернет и мобильных средств связи, благодаря которым пассажиры и водители, ранее не знакомые, смогли быстро находить друг друга и договариваться о деталях поездки. В наши дни существуют следующие схемы карпула [1].

**Поиск через форумы и сообщества.** Такими ресурсами являются, например, «Давай со мной» [3], «Довежу!ру» [4], eRideShare.com [5], PickupPal [6], Zimride [7], RideshareOnline [8], rideshare.511.org [9], CarJungle [10]. На сайте размещается объявление о поездке, которое содержит в себе информацию о человеке, подающем это объявление, и маршрут с указанием начальной и конечной точек. Поиск попутчиков осуществляется заданием маршрута поездки и дополнительных условий, таких как плата за проезд, время поездки и др.

**Поиск через закрытые веб-сайты.** Закрытость выражается в необходимости приглашения для участия. Например, сервис Zimride [7] вместе с общедоступным интерфейсом имеет закрытый, который могут использовать различные организации для обеспечения подвоза своих сотрудников к месту работы.

**Использование специализированного программного обеспечения для поиска.** На мобильные устройства загружается программа, которая позволяет отредактировать профиль пользователя, указать маршрут и найти попутчиков. Примеры: PickupPal [6], Avego [11].

**Поиск через различных агентов (таксомоторные компании).**

**Случайный подбор попутчиков на точках встречи,** которыми могут быть, например, остановки общественного транспорта.

Уже существующее программное обеспечение для мобильных устройств поддерживает клиент-серверную архитектуру, реализация которой пред-

усматривает наличие централизованного сервера и клиентов, осуществляющих запросы на обработку данных к серверу.

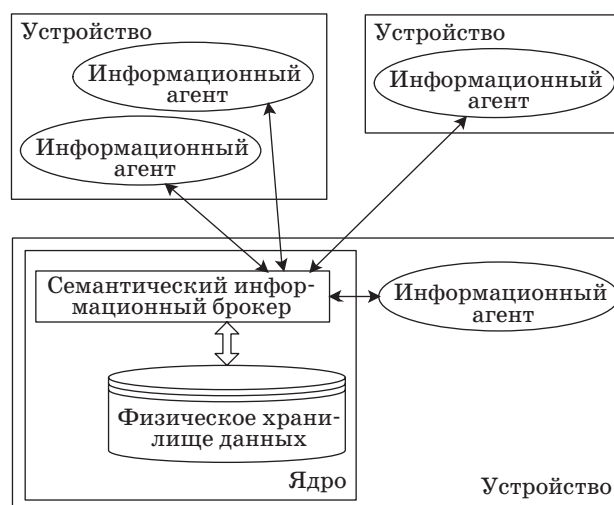
Настоящая работа предусматривает децентрализацию вычислительного комплекса, которая осуществляется использованием технологий Semantic Web и, в частности, интеллектуальных пространств (Smart Spaces) на основе платформы с открытым исходным кодом Smart-M3, разрабатываемой Nokia Research Center [12, 13].

### Платформа Smart-M3

Платформа Smart-M3 объединяет в себе идеи распределенных сетевых систем и Semantic Web. Ее ключевыми идеями являются независимость от конкретных производителей, оборудования, области применения и возможность обмена информацией между различными программными модулями посредством простого и общедоступного информационного брокера. Благодаря использованию Semantic Web в качестве базы, обмен информацией между участниками пространств может осуществляться на основе протокола HTTP и с использованием унифицированных идентификаторов ресурсов (Uniform Resource Identifier — URI) [14].

Общая структура платформы представлена на рис. 1. Ядро системы подразделяется на 2 элемента: СИБ (семантический информационный брокер, Semantic Information Broker — SIB) и физическое хранилище данных.

Семантический информационный брокер предоставляет доступ информационным агентам к информационному пространству, обеспечивая их функциями обработки информации, такими как вставка, извлечение, редактирование, удаление и подписки на изменение информации в ин-



■ Рис. 1. Структура платформы Smart-M3

теллектуальном пространстве. В хранилище данных вся информация сохраняется как граф, удовлетворяющий требованиям стандарта RDF (Resource Description Framework) [15], описывающего семантические сети, в которых узлы и дуги имеют унифицированные идентификаторы ресурсов. Каждое утверждение в соответствии с этим стандартом описывается тройкой «субъект — предикат — объект», например, «Иван — имеет — автомобиль», и по своей сути является простым предложением. Информационные агенты — это программные модули, с помощью которых осуществляется взаимодействие с семантическим информационным брокером через Smart Space Access Protocol (SSAP — протокол доступа к интеллектуальному пространству) [13].

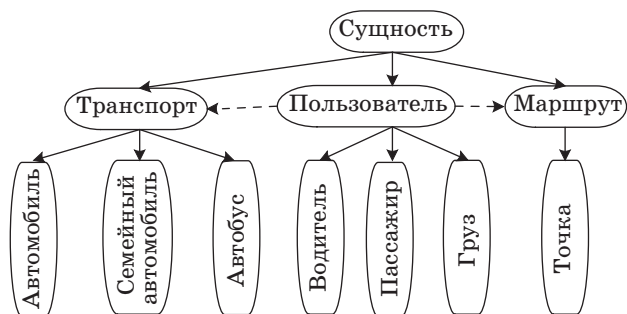
### Онтология системы

В ходе разработки системы была спроектирована ее онтология (рис. 2).

В системе присутствует три типа сущностей: транспорт, пользователи и точки маршрута. Транспорт — это обычные автомобили, в которых число мест для пассажиров не больше четырех, семейные автомобили с числом мест от пяти до восьми и автобусы, имеющие девять и более мест для пассажиров.

«Точка» является частью маршрута и используется для определения путей пользователей. Ее онтология представлена на рис. 3.

Пользователями в системе выступают водители, пассажиры и грузы. При этом все они имеют связь с транспортом и точками. Так, например, водитель является собственником транспорта, и ему соответствуют несколько точек, определяющих положение дома, работы, мест остановок и т. п. Пассажир может отдавать предпочтение определенным видам транспортного средства, и ему также соответствуют точки на карте, характеризующие его положение и пункт назначения. Груз имеет определенные размеры, и не каждый тип транспорта может подойти для его перевозки.



■ Рис. 2. Онтология логистической системы на макроуровне

Для описания путей пользователей используется множество точек, являющихся экземплярами сущности «точка». Этот набор представляет собой упорядоченный список опорных точек, полученных с помощью алгоритмов нахождения кратчайшего пути на графе (например, алгоритма Дейкстры или Флойда—Уоршелла), «натянута» на топографический план местности. Вершины графа при этом являются опорными точками, а дуги повторяют положение и направление основных дорог. Фрейм «точка» представлен на рис. 3.

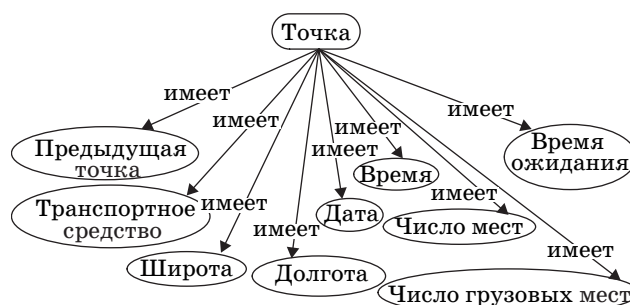
В составленной онтологии фрейм «точка» содержит в себе следующие атрибуты:

- предыдущая точка пути. Имеет значение false, если точка является начальной;
- широта точки пути;
- долгота точки пути;
- транспортное средство. В случае, если рассматриваемая точка является точкой пути пассажира, она указывает на водителя, который в данный момент подвозит этого пассажира. Если пассажир идет пешком, имеет значение false;
- число свободных мест для пассажиров в транспортном средстве в данной точке;
- число свободных мест для груза в транспортном средстве в данной точке;
- расчетная дата нахождения пользователя в данной точке;
- расчетное время нахождения пользователя в данной точке;
- время, в течение которого пользователь должен находиться в данной точке.

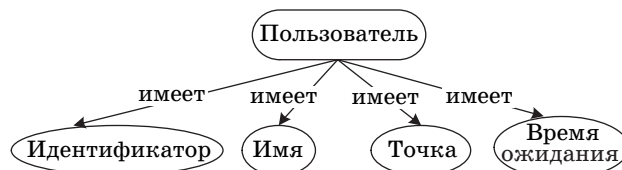
Описание пользователя представлено на рис. 4.

Основными атрибутами для всех пользователей являются:

- уникальный идентификатор пользователя;



■ Рис. 3. Фрейм «точка пути»



■ Рис. 4. Фрейм «пользователь»

- имя пользователя;
- точки, соответствующие пользователю (как минимум две: начальная и конечная);
- время, в течение которого данный пользователь согласен ждать других пользователей.

Пассажиры, водители и грузы наследуются от сущности «пользователь». Сущность «водитель» наследует все атрибуты «пользователя», при этом расширяя его еще двумя атрибутами:

- vehicle — тип транспортного средства;
- detour — максимально возможное отклонение от кратчайшего маршрута.

Сущность «пассажир», так же, как и «водитель», наследует все атрибуты «пользователя» и добавляет к ним атрибут detour, аналогичный подобному в сущности «водитель».

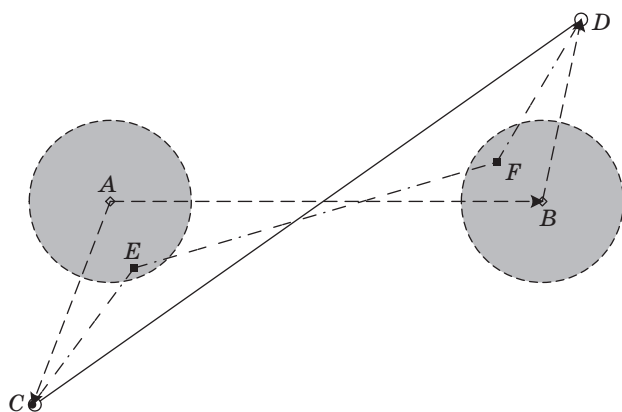
Сущность «груз» помимо атрибутов сущности «пользователь» имеет атрибут size, определяющий размеры груза.

### Алгоритм поиска совпадающих путей и точек встречи водителя и пассажира

Для поиска совпадающих путей составляются списки водителей и пассажиров, при этом для каждой пары ищутся совпадения. Из полученных пар выбираются пары, наиболее удовлетворяющие критериям, заданным участниками.

Общий принцип поиска совпадающего пути рассмотрен на рис. 5.

Пусть  $A, B$  — начальная и конечная точки пути пешехода,  $C, D$  — начальная и конечная точки пути водителя. Сплошной линией обозначен кратчайший путь водителя, полученный с помощью ГИС. Как видно из рисунка, водитель и пешеход осуществляют движение практически в одном направлении, и при некоторой корректировке пути водителя он может подвезти пешехода, что обозначено на рисунке пунктирной линией (путь  $CABD$ ). Рассмотренный случай является самым простым, так как точки встречи в нем со-



■ Рис. 5. Общий принцип поиска совпадающего пути

впадают с точками начала и конца пути пешехода. Более сложными вариантами является поиск точек встречи, удовлетворяющих и водителя, и пассажира, но не обязательно являющихся частью их кратчайшего пути. На рисунке один из возможных вариантов обозначен штрихпунктирной линией, а точки встречи обозначены буквами  $E, F$  (путь  $CEFD$ ). Выбор этих точек должен удовлетворять следующим основным условиям:

- расстояние от начальной точки пассажира до точки встречи не должно превышать максимального пути, который согласен пройти пассажир. На рис. 5 эта область обозначена пунктирной окружностью;
- отклонение нового пути водителя от кратчайшего не должно превышать максимально допустимого отклонения, задаваемого водителем.

Основная схема алгоритма поиска совпадающих путей будет выглядеть следующим образом:

```
FOR EACH driver DO
  FOR EACH passenger Do
    Find_matching_path(driver.path,passenger.path); // Согласно
    представленным выше условиям.
    constraint_checking();
    IF ALL constraints IS performed THEN
      set_passenger_for_driver();
    ENDFOR;
  ENDFOR;
```

Общая задача поиска совпадающих путей имеет достаточно большую размерность, поэтому необходимо применять эвристики для уменьшения размерности задачи.

Перед работой алгоритма целесообразно отсеять водителей, которые априорно не имеют возможности подвезти пассажира. Для такого отсеивания необходима проверка выполнения следующих условий:

$$(pp_1^x - dp_i^x)^2 + (pp_1^y - dp_i^y)^2 \leq (Pdetour + Ddetour)^2; \quad (1)$$

$$(pp_2^x - dp_i^x)^2 + (pp_2^y - dp_i^y)^2 \leq (Pdetour + Ddetour)^2, \quad (2)$$

где  $pp_1, pp_2$  — начальная и конечная точки пути пассажира;  $dp_i$  — точка пути водителя;  $Pdetour, Ddetour$  — отклонение от минимального пути пассажира и водителя. Если путь водителя не удовлетворяет условиям (1), (2), то считается, что водитель не может подвезти пассажира, иначе составляются списки точек пути водителя, соответствующих условиям, и осуществляется переход к шагу, на котором производится выбор всех возможных точек встречи.

Следующим шагом производится выбор всех точек, до которых может дойти пешеход. Эти точки принадлежат окружности радиуса  $Pdetour$  и выбираются по следующему правилу:

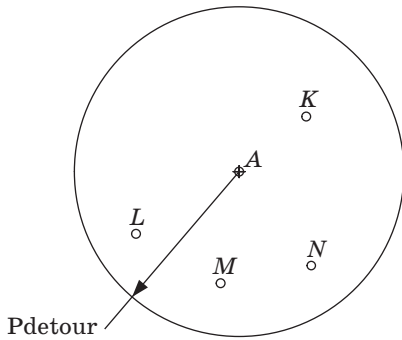


$$(pp_A^x - p_i^x)^2 + (pp_A^y - p_i^y)^2 \leq Pdetour^2, \quad (3)$$

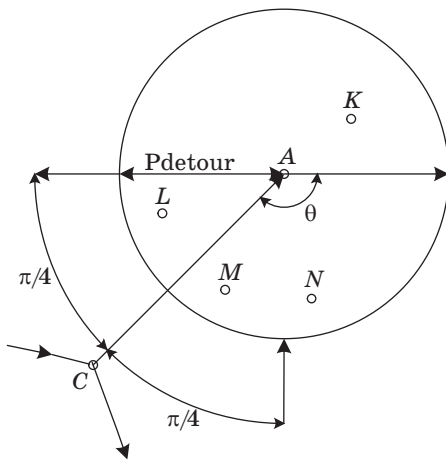
где  $pp_A$  — точка, принадлежащая пути пассажира;  $p_i$  — точка на карте. Если  $p_i$  удовлетворяет условию (3), то она помечается как возможная точка встречи. На рис. 6 точки, удовлетворяющие условию (3), обозначены буквами  $L, M, N, K$ .

Выбор из возможных точек встречи пары, наиболее удовлетворяющей критериям, заданным пассажиром, предусматривает перебор всех пар точек с вычислением всех требуемых для выбора параметров. Данная операция является крайне затратной и требует сокращения множества возможных точек встречи. Для решения поставленной задачи были разработаны эвристики, позволяющие на основе имеющихся данных сократить множества.

Первая эвристика предполагает выбор точек из сектора окружности, со стороны которой будет ехать водитель. На рис. 7 представлена эвристика для ситуации, в которой у водителя нашлась только одна точка (точка  $C$ ), удовлетворяющая условию (1) или (2), в зависимости от рассматриваемой точки пути пассажира.



■ Рис. 6. Выбор точек встречи для пассажира



■ Рис. 7. Эвристика в ситуации, когда в пути водителя нашлась только одна допустимая точка

Чтобы определить потенциальные точки встречи, вычисляется угол

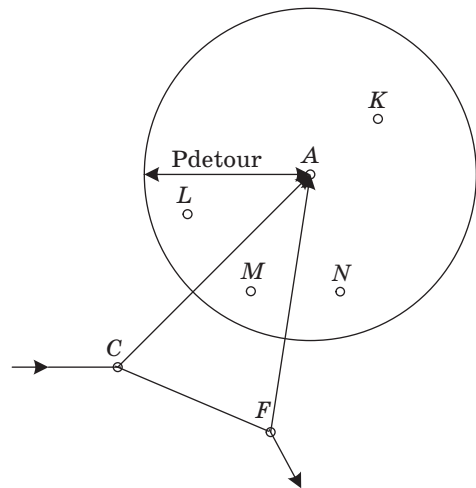
$$\theta = \arctg\left(\frac{C^y - A^y}{C^x - A^x}\right) \quad (4)$$

и выбираются точки, которые попадают в область  $\left[\theta - \frac{\pi}{4}, \theta + \frac{\pi}{4}\right]$ . На рис. 7 это точки  $L$  и  $M$ . Точка  $A$ , являющаяся начальной точкой для пассажира, всегда будет включена в множество возможных точек встречи. Если нашлись две и более точки, удовлетворяющие условию (1), то область выбора точек расширяется. На рис. 8 представлена схема применения эвристики, полученной из схемы, изображенной на рис. 7, путем добавления второй точки, удовлетворяющей условию (1) или (2). При этом видно, что в расширенную область вместе с точками  $L$  и  $M$  теперь входит и точка  $N$ .

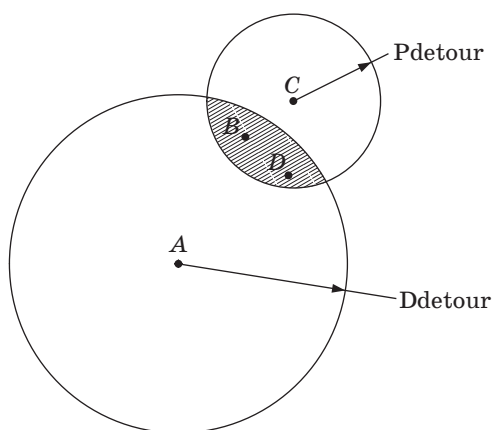
Применение подобной эвристики позволяет рассматривать только точки, находящиеся со стороны движения водителя. Однако она обладает существенными недостатками:

- выбор точек, расположенных дальше, чем максимальное отклонение водителя от кратчайшего пути;
- область выбора, ограниченная определенным углом. Из-за некорректного выбора угла возможна потеря потенциальных точек встречи.

Схема применения второй эвристики представлена на рис. 9. Суть второй эвристики заключается в том, что осуществляется выбор точек, находящихся на пересечении окружностей радиусов  $Pdetour$  и  $Ddetour$ . За счет этого все точки являются потенциально достижимыми как для водителя, так и для пассажира, при этом отпадает необходимость определения угла, ограничива-



■ Рис. 8. Схема для двух точек пути водителя



■ Рис. 9. Схема применения второй эвристики

ющего область выбора. Как и в первой эвристике, рассматриваемая область может расширяться за счет увеличения количества точек пути водителя, удовлетворяющих условию (1) или (2).

Для эффективной работы представленных эвристик необходимо выполнение следующих условий:

- большое количество водителей. Если водителей будет мало, то из-за «жесткого» отсева поиск точек будет редко давать положительный результат;

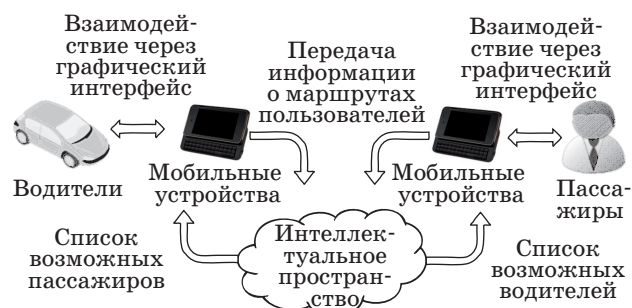
- малое значение  $Ddetour$ . При больших значениях  $Ddetour$  эвристика будет бесполезной, так как задача сведется к полному перебору;

- равномерное распределение дорог по карте. Наличие рек, озер, заводов и т. п. ведет к отсутствию дорог в некоторых секторах (что является причиной потери возможных точек встречи), связанному с необходимостью объезда препятствия и заезда за пешеходом с другой стороны при достаточно больших значениях  $Ddetour$ .

### Сценарий работы системы

Общий сценарий функционирования системы представлен на рис. 10.

Все участники системы устанавливают специальное приложение на мобильное устройство.



■ Рис. 10. Сценарий функционирования системы

С согласия пользователя приложение собирает информацию о расписании пользователя, личных предпочтениях, наиболее частых маршрутах и т. д. Также пользователь может выставлять дополнительные ограничения, например, максимально допустимую задержку в пути, максимальное дополнительное расстояние, которое водитель готов проехать в интересах пассажира, интересы для социального общения и т. п. После внутренней обработки и деперсонализации информации (передается не сама информация, а ее признаки) она передается в интеллектуальное пространство для формирования групп наиболее подходящих попутчиков в рамках собранных ограничений. Группы формируются в ходе исполнения логистического алгоритма, представленного выше. Пользователи из одной группы интерактивно получают список возможных попутчиков с описанием их профилей и, если разрешено пользователем, со ссылкой на внешние ресурсы, например страницу социальной сети (Facebook), точки и время встречи, и полные рекомендации по маршруту. Кроме предварительного поиска попутчиков возможен режим постоянного поиска в режиме реального времени в зависимости от изменения контекстной информации, например, при подключении к пространству новых пользователей, внесении изменений в ограничения и т. д. Таким образом, от пользователя не требуется никаких действий для поиска подходящих попутчиков, вся работа выполняется на стороне интеллектуального пространства, при этом особое внимание уделяется сохранению ощущения личного пространства у пользователя.

### Заключение

В работе представлена интеллектуальная система управления логистической сетью при совместном использовании автотранспорта, позволяющая находить попутчиков для водителей в режиме реального времени с использованием персональных мобильных устройств. Система основывается на информационной платформе Smart-M3, реализующей концепцию интеллектуального пространства. Использование этой платформы позволяет значительно повысить масштабируемость и расширяемость прототипа системы. Представлен алгоритм поиска попутчиков для водителей и определения их точек встречи. В рамках алгоритма были разработаны две эвристики, что обеспечило ускорение его работы более чем в 1,5 раза.

Совместное использование транспорта при массовом применении системы существенно снизит нагрузку на транспортные системы городов, уменьшит дорожные заторы, обеспечит эконо-

мию топлива, снизит расходы на парковку и т. д. Например, согласно исследованиям Массачусетского технологического института, совместные поездки позволяют снизить суммарный пробег автомобилей на 9–27 % ежедневно [16].

В публикации представлены результаты исследований, инициированные контрактом с Nokia Research Center в области интеллектуальной логистики (Smart Logistic), выполненные при поддержке грантом РФФИ 10-07-00368-а, програм-

мой Президиума РАН «Интеллектуальные информационные технологии, математическое моделирование, системный анализ и автоматизация» (проект 213) и программой Отделения нанотехнологий и информационных технологий РАН «Интеллектуальные информационные технологии, системный анализ и автоматизация (проект 2.2 «Разработка методологии построения групповых информационно-рекомендующих систем»).

## Литература

1. **Carpool** // Wikipedia. <http://en.wikipedia.org/wiki/Carpool> (дата обращения: 09.07.2012).
2. **Smirnov A., Kashevnik A., Shilov N. et al.** On-the-Fly Ontology Matching in Smart Spaces: A Multi-Model Approach. In: Smart Spaces and Next Generation Wired/Wireless Networking: Proc. of the Third Conf. on Smart Spaces, ruSMART 2010, and 10<sup>th</sup> Intern. Conf. NEW2AN 2010, St. Petersburg, Russia. Springer, 2010. LNCS 6294. P. 72–83.
3. **Давай со мной.** <http://www.gowithme.ru/> (дата обращения: 09.07.2012).
4. **Довезу!ру.** <http://www.dovezu.ru/> (дата обращения: 09.07.2012).
5. **eRideShare.com.** <http://erideshare.com/> (дата обращения: 09.07.2012).
6. **PickupPal.** <http://www.pickupal.com/> (дата обращения: 09.07.2012).
7. **Zimride.** <http://www.zimride.com/> (дата обращения: 09.07.2012).
8. **RideshareOnline.** <http://www.rideshareonline.com/> (дата обращения: 09.07.2012)
9. **Rideshare 511.** <http://rideshare.511.org/> (дата обращения: 09.07.2012)
10. **CarJungle.** <http://www.carjungle.ru/> (дата обращения: 09.07.2012)
11. **Avego.** <http://www.avego.com/> (дата обращения: 09.07.2012)
12. **Honkola J., Laine H., Brown R., Tyrkko O.** Smart-M3 Information Sharing Platform: Proc. IEEE Symp. Computers and Communications (ISCC'10) / IEEE Comp. Soc., 2010. P. 1041–1046.
13. **Smart-M3** // Wikipedia. <http://en.wikipedia.org/wiki/Smart-M3> (дата обращения: 19.02.2011)
14. **Berners-Lee T., Fielding R., Masinter L.** RFC 3986 — Uniform Resource Identifier (URI): Generic Syntax // Дата обновления: январь 2005. <http://tools.ietf.org/html/rfc3986> (дата обращения: 09.07.2012)
15. **Resource Description Framework (RDF).** <http://www.w3.org/RDF/> (дата обращения: 09.07.2012)
16. **Rideshare History & Statistics.** MIT «Real-Time» Rideshare Research. <http://ridesharechoices.scripts.mit.edu/home/histstats> (дата обращения 09.07.2012).