ARTICLE IN PRESS

Transportation Research Part C xxx (2012) xxx–xxx

Contents lists available at SciVerse ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

The TranQuyl language for data management in intelligent transportation ☆

Ouri Wolfson *, A. Prasad Sistla [1], Bo Xu

Department of Computer Science, University of Illinois at Chicago, United States

ARTICLE INFO

Article history:
Received 18 January 2012
Received in revised form 1 February 2012
Accepted 1 February 2012
Available online xxxx

Keywords:
Intelligent Transportation Systems
Multiple modality
Information search
Query processing
Spatio-temporal data management

ABSTRACT

Intelligent Transportation Systems envision a networked environment consisting of vehicles, the infrastructure, and hand-held devices (e.g., smart-phones). The environment will enable numerous safety, mobility, and environmental improvement applications. For example, drivers can be warned of dangers in their local environment or when risking to leave their lane. Furthermore, their visibility range can be expanded by providing highly up-to-date information from areas that are currently invisible. For another example, the road weather—up-to-the-minute visibility, precipitation, and pavement condition information—can be provided at high spatial resolution.

Intelligent Transportation efforts are currently being undertaken throughout the world. In addition to the IntelliDrive initiative of the US Department of Transportation, similar efforts exist in Europe, Japan, and China. But these efforts are largely decoupled from, and often incognizant of, the advances in spatio-temporal information management.

This paper outlines a spatio-temporal data management language, Transportation Query Language (TranQuyl), which will facilitate the specification of a wide variety of queries of interest to travelers, to transportation agencies, and to industry. Queries in TranQuyl may be processed in either client server mode, or mobile peer-to-peer (P2P) mode, or both. TranQuyl will provide support for the specification of uncertainty either quantitatively or qualitatively as fuzzy queries, for example: "retrieve safety/emergency information around me". In response, query processing should avoid overloading the traveler with information, and instead present only the most relevant answers to the query.

1. Introduction

The impact of Computer Science (CS) and Information Technology (IT) on transportation systems is not as dramatic as the one on science, finance or business in general. But in the last few years we have witnessed significant penetration of IT in surface transportation. Navigation systems with real-time traffic information and route planning capability, color coded traffic maps, and real-time information displays about public transport vehicles (e.g. nextbus and CTA's bus-tracker) are some examples of the improvements in urban transportation brought by IT. The rapid advances in mobile and ubiquitous computing and sensor networks are opening opportunities to revolutionize large complex systems, including transportation. Indeed, the purpose of the IntelliDrive[SM] initiative of the US Department of Transportation is "advancing connectivity among vehicles and roadway infrastructure in order to significantly improve the safety and mobility of the US transportation system" (RITA, 2010). A related development is the emergence of increasingly more sophisticated geospatial and temporal information management capabilities. These factors have the potential to dramatically alter traveler services, and the provision and analysis of related information.

☆ Research supported by NSF DGE-0549489, 0916438 and CNS-1035914.

* Corresponding author. Tel.: +1 312 996 6770; fax: +1 312 413 0024.
E-mail addresses: wolfson@cs.uic.edu (O. Wolfson), sistla@uic.edu (A.P. Sistla), boxu@cs.uic.edu (B. Xu).
[1] A. Prasad Sistla is also an adjunct professor at King Abdulaziz University.

0968-090X/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
doi:10.1016/j.trc.2012.02.002

In the envisioned environment, travelers and billions of sensors, both embedded within the infrastructure and in vehicles and smartphones, will generate vast amounts of data whose interpretation could be exploited to spur the formulation of innovative transportation services and policies. Advances in social networking, crowd-sourcing, and data mining research are increasingly creating new sophisticated mechanisms which can foster seamless information integration among travelers, provide alternatives, and support sustainable economic and social policies.

In addition to technological advances, novel applications are driven by the following factors (Winter et al., 2011):

- Ever increasing mobility demand leading to congestion with dramatic effects on public safety, on the environment, and on the economy.
- Real infrastructure is expensive and laborious to build and maintain; furthermore, it is aging and has to be replaced by modern, new concepts and systems.

Most novel applications are included in the following classes (Winter et al., 2011):

1. Shared transportation resources: a better exploitation of the resources is achieved by sharing, with benefits for the users (reduced prices), the infrastructure (less congestion), and as a consequence also the environment (less pollution).
2. Collaborative traveling, for example by platooning, i.e. the virtual coupling of vehicles to form larger units like virtual trains. These structures can get priorities e.g. when crossing junctions. Within a platoon, autonomous driving is possible. Also, intelligent traffic lights enable a more adaptive giving right of way depending on the current traffic situation instead of fixed schedules.
3. Infrastructure is replaced by virtual infrastructure: in this way, the real infrastructure, which has several disadvantages like aging and expensive maintenance, can be replaced. Examples are virtual traffic lights where vehicles negotiate right-of-way, virtual signs; it is also relevant for highly temporal and ad-hoc warnings like construction sites or aquaplaning or slippery roads.
4. Driver assistance: drivers can be warned of risks in their local environment or when risking to leave their lane. Furthermore, their visibility range can be expanded by providing highly up-to date information from areas that are currently invisible.
5. Evacuation planning: highly temporal information is provided to support and calibrate simulations
6. Autonomous driving: as a long-term goal, highly dynamic maps of the environment have the potential to support autonomous driving.
7. Dynamic road pricing: the knowledge about the current usage of roads can be used to manage traffic, e.g. by reducing prices for collaboratively used cars or platoons.
8. Smart grid, electric cars: sharing resources opens the way to extend the flexibility of using and sharing electric cars, e.g. by dynamic planning of the electric grid resources, and of routes by considering charging facilities.
9. Road and traffic planning can be greatly enhanced by precise, high resolution travel information, which leads to adaptive traffic systems. For example, the road weather—up-to-the-minute visibility, precipitation, and pavement condition information—can be provided at high spatial resolution.

Indeed Intelligent Transportation efforts are currently being undertaken throughout the world. In addition to the IntelliDrive initiative of the US Department of Transportation (RITA, 2010) mentioned above, similar efforts exist in Europe, Japan, and China. But these efforts are largely decoupled from, and often incognizant of, the advances in spatio-temporal information management and ubiquitous sensing. Indeed, the potential of crowd-sourcing through the billions of mobile sensors in vehicles and smartphones that currently exist in the transportation system is largely untapped. But things are starting to change in the sense that the Civil Engineering community, which is driving the Intelligent Transportation efforts, becomes aware of the potential of spatio-temporal information tools to facilitate the large scale deployment of Intelligent Transportation applications.

Towards this end, i.e. facilitating Intelligent Transportation applications, we propose three objectives, corresponding to query-language design, query-processing, and answer filtering.

1.1. Language and tools for data management

In the Intelligent Transportation environment there will be a large amount of data being collected and stored. This data will typically be collected in a distributed network (e.g., traffic sensor network or probe vehicles) and stored locally or on a central server. The data will then be disseminated through peer-to-peer, client–server, or web-based mechanisms and may be accessed in real-time. In other words, there will be numerous distributed data sources co-existing in the network. This requires a novel database management system (DBMS) that effectively identifies the data sources and integrates the queried data from these sources. As part of the DBMS platform, we need to develop data models for storing such data, user friendly languages for querying the data, and tools for efficient processing of the queries. The data model should be spatio-temporal, storing moving object data as well as graph based data that captures the transportation network and its facilities. The system should provide language mechanisms for querying all such data including trips in a multi-modal (e.g. car, bus, and train) transportation network. It should allow queries about available resources, traffic conditions, dynamic route planning, etc.

In other words, it should allow queries arising in the novel application classes discussed above. We propose the Transportation Query Language (TranQuyL) for this purpose.

1.2. Mobile P2P query processing

In many applications, due to performance, cost, and privacy considerations, a query is better be processed in a mobile P2P network rather than in client–server mode; or in a combined (client–server and mobile P2P) mode. The mobile P2P network may involve communication among vehicles (V2V communication) and communication between vehicles and portable devices such as smart-phones. Such processing is particularly challenging in highly dynamic P2P environments. In other words, we need to develop efficient query processing algorithms for TranQuylin mobile P2P environments. We carried out initial work in this direction in (Ma et al., 2010; Xu et al., 2009, 2010).

1.3. Answer filtering and ranking by relevance

TranQuyl queries often access data generated by sensors, and there is a mismatch between the queries posed by travelers and sensor data. In other words, travelers pose queries at a higher level of abstraction than that generated by sensors. This means thatTranQuyl queries are often fuzzy. For example, a driver D is always interested in safety related information (a fuzzy query Q), e.g. an emergency-braking vehicle ahead. Clearly information such as emergency braking of a vehicle V can be generated automatically by the brake-sensors of V. But is this information relevant to D, i.e. does it properly answer Q? For example, is it relevant if V is20 meters ahead of D? Probably so. What about 2 miles ahead? Probably not. So it is unclear which sensor-generated data answers Q. Similarly D may be interested in parking information. Is information about an available parking slot 20 meters away relevant, i.e. likely to still be available when D reaches it? Probably so. What about a parking slot 200 m away? The downside of presenting to the traveler all the possible answers is information overload, which in turn may desensitize and distract the traveler, causing the information provided to be ignored and introducing safety hazards. We propose to address the mismatch by ranking answers to fuzzy queries according to relevance, and presenting only the most relevant.

In Sections 2–4 we elaborate on each one of the three objectives, respectively. Each section, discusses preliminary results, and the research issues associated with an objective. In Section 5 we discuss relevant work.

2. Language and tools for querying and management of data

As indicated, one of the main aims of the paper is to propose a DBMS framework in support of ITS applications. The DBMS must be able to query distributed data stored on web sites, various mobile vehicles, and on central servers, and be able to answer various queries on such data. The queries can be—"Find a route that will get me home by my designated time with at least 90% certainty;" or "Using public transportation, find a route that lets me stop at a grocery store for 30 min and reach home by 7:00 pm." Such queries pertain to routes in multi-modal transportation networks (i.e. a network with multiple modes of transportation) involving various constraints, optimization criteria together with uncertainty clauses. Other types of queries can be about traffic conditions or about available resources, etc. For example, a typical query issued by a driver on a highway is "what is the average speed one mile ahead of me?" or "what are the available parking spots within two blocks of my current location?" There can also be queries about potholes on a route. Some queries may pertain to safety hazards such as an ice patch, a malfunctioning brake-light, or an emergency brake ahead.

A query such as the one about the speed one mile ahead can be posed to a central server, but sometimes this information is unavailable on the server (e.g. because the query pertains to a congested side street that is not instrumented with speed sensors), and the query needs to be answered by polling the vehicles ahead. However, the network id's of these nodes are not known. Thus, for a query it may not be known where the data resides, and how to get to it. Of course, the answer in this case is to use short-range wireless communication such as Wifi or DSRC (Carter, 2005)to disseminate the query to neighboring nodes transitively. In other words, the limited transmission range is used to compensate for the lack of id's knowledge.

These issues are not addressed by traditional DBMS's. The data integration problem in database literature assumes that the data is always available, and the integration part is the problematic one. In distributed databases it is assumed that there are directories that map data to network id's of computers that store the data. Such directories are appropriate for some of the data that pertains to the query, but certainly not to queries that are processed by polling other vehicles.

First we propose a data model for storing and querying data in multi-modal transportation networks. Preliminary work using this approach can be found in (Booth et al., 2009). Then we discuss proposed research on this topic. We also propose additional research about maintaining information and querying dynamic information about traffic conditions, resources and network status as posed above.

2.1. Multimodal route planner in urban transportation networks

In recent years we have seen a growing number of resources that provide online maps, directories, location-based services, and route planners that attempt to bring the necessary information to users. However, these systems do not provide a comprehensive solution to transportation information systems. Part of the difficulty in developing fully integrated systems is the heterogeneous nature of the data and the lack of a coherent data model that can be used effectively. In this proposal, we propose a method to integrate the key aspects of spatio-temporal, moving objects, and graph-based databases.

A typical query about route planning is one of the above queries about finding a route from home to a destination satisfying some time constraints, some optimization criteria and satisfying some uncertainty constraints. In these queries we see that routes, or trips, are the primary focus. These trips have spatial and temporal constraints and are subject to uncertainties. The goal is to provide a powerful and easy to use system to pose such queries based on trips and to process such queries efficiently.

Our proposed work on trip planning and dynamic routing in multimodal systems is based on a graph model of the network. This graph model contains not only the topology of the transportation network, possibly at the lane level, but also various other information such as real-time traffic information including predicted travel times, dynamic schedule information, available facilities and hazards. The trip planning and dynamic routing system provides a user friendly language for querying trips to desired destinations with requirements on intermediate facilities to be visited, safety related information, desired constraints and optimizing on different criteria such as travel time and cost. It also allows specification of certainty requirements in quantitative as well as qualitative terms. Thus, the proposed system goes way beyond any such currently available system.

The remainder of the section is organized as follows: first (Section 2.2) we introduce a multimodal urban transportation network (Section 2.2.1), its representation as a graph (Section 2.2.2), and the initial version of TranQuyl (Section 2.2.3) for routing in multi modal transportation networks. Then, in Section 2.3 we discuss the required future work.

2.2. Technical approach

2.2.1. Urban transportation networks

The urban transportation network has both static and dynamic components. The physical structure of the network does not change—buildings, roads, lakes, and train tracks remain constant (ignoring long term construction). However, the position of people and vehicles, status of the roads, departure times of buses, and similar components change continuously in real time.

The transportation network itself is composed of numerous routes that correspond to some physical paths. The most obvious paths are roads that can carry automobiles, buses, and in some cases pedestrians. There are also railroad tracks that carry trains. These routes are all labeled with unique identifiers (e.g., Red Line train, #12 bus, Roosevelt Rd., Interstate 94). It is possible for the routes to overlap on the same physical path (e.g., the #12 and #15 buses run on Roosevelt Rd.).

A key feature in urban transportation is the presence of multiple modes of transportation. The modes that we consider in our work are auto, pedestrian/bicycle, bus, and rail. It is important to note that pedestrian is a specific mode of transportation. We do not consider air, water, or long range intercity travel at this time. These modes could likely be modeled in the same manner, but are beyond the scope of the current proposal.

We introduce the concept of a trip, which is a path from an origin to a destination through the network. In addition to the specification of origin and destination, we allow specification of other types of constraints be placed on a trip, including: desired modes, facilities, path constraints, time constraints and uncertainty constraints.

Facilities are resources such as gas stations and electric cars charging stations, banks, ATMs, and grocery stores that users may need to include on a trip. Path constraints require the trip to avoid or include some regions or (possibly high-resolution) weather conditions. Time constraints require the duration or departure or arrival times to satisfy certain conditions. In addition, we allow specification of optimization criteria on distance or time duration, or number of intermodal transfers, etc. We also allow specification of certainty measures. Trips are subject to some level of uncertainty due to the dynamic and unpredictable nature of transportation systems. We assume that we have knowledge of the current speeds on links in the network as well as the expected arrival and departure times for public transportation. Being a real-world system, this information is unreliable; therefore, we describe it probabilistically. We allow specification of a quantitative measure of the certainty of the result generated by a query.

2.2.2. Graph model

We begin by defining the graph model used for representing the transportation network for querying trips. We define a transportation network to be a tuple U = (M, F, L, G). M is the set of modes available in the network. The set F represents the classes of facilities (e.g., electric cars charging station, grocery store, fast food restaurant, gas station) available on the transportation network. The set L denotes the attributes (e.g., length, name, mean speed) for the edges in the network. G, the network, is a labeled, directed, multi-graph. Note that we call G as a multi-graph since it can have multiple-edges between the same pair of nodes denoting multiple ways of traveling between the two nodes.

For edges, we have a set of attributes which may differ for different modes. Some edge attributes are name, mean speed, geometry and speed variance. Each vertex also has a set of vertex attributes which are name, geometry and facilities.

We define the graph as G = (V, E, f) where V is the set of labeled vertices and E is a set of labeled edges. Each edge is a 4-tuple (u, v, m, g) where u, v are the endpoints of the edge, m is the mode label of the edge, and g is a function that maps the attributes defined for the mode to values. For example, for an edge with mode rail, we have attributes such as run_id and departure_time denoting the train number and departure time from the source vertex of the edge. In other words, trains

Please cite this article in press as: Wolfson, O., et al. The TranQuyl language for data management in intelligent transportation. Transport. Res. Part C (2012), doi:10.1016/j.trc.2012.02.002

starting at different times on the same route are captured by parallel edges having different run_id's and departure times; similarly for buses. Thus G captures the network and schedules information in a unified framework.

For each vertex v and vertex attribute x, f specifies the value of attribute x for vertex v. Edges of multiple modes may be incident on the same vertex, and in fact this is how the transfer between modes is modeled. Furthermore, edges of different modes (e.g. bicycle and bus) may be parallel if both modes are allowed on a road section. For a vertex, the value of the attribute Facilities specifies the facilities available at that vertex. For example, aflorist is available at a station vertex.

We define a leg to be a sequence of alternating vertices and edges starting and ending with a vertex where all of the edges have the same name (e.g. Bus line 15), mode, and if available, run_id. For each edge in the leg, its start vertex is same as the vertex preceding the edge in the leg; the end vertex of an edge is same as the vertex following the edge in the leg. We define the departure time of a leg to be the departure time of its first edge; similarly, we define its arrival time to be the arrival time of its last edge. A trip is a sequence of legs, where the beginning vertex of each successive leg in the sequence is same as the end vertex of the preceding leg, such that the departure time of each subsequent leg is greater than or equal to the arrival time of the previous leg. We define the departure time and arrival time of a trip to be the departure time and arrival time of the first and last leg respectively.

We define a transfer to be a vertex shared by two different legs in the same trip. The transfer is intermodal if the modes of the two legs are different, and intramodal if they are the same.

Fig. 1 shows an example graph of a transportation network.

2.2.3. The relational network model and query language

In this subsection we define the Transportation Query Language (TranQuyL) for specifying queries on our graph model. The language focuses on querying trips subject to various constraints—including that of uncertainty. Our query structure builds on the standard "select, from, where" structure of SQL. We retain the same base syntax and structure but extend it with additional clauses. First, to query trips we introduce the operator ALL_TRIPS (origin, destination) that accepts two vertices origin and destination as arguments. It returns the set of all trips from the source to the destination in the graph. A generic query has the structure shown in Fig. 2.

The above query selects those trips from the origin to destination that satisfy the "where" condition and that satisfy the criteria specified by other clauses. The "with stop_vertices" clause specifies the vertices where stops are required. The "with modes" clause specifies the modes that are allowed. As indicated above, the "where" clause specifies conditions on the stop vertices and any other conditions. The "with certainty" clause requires the certainty that is desired as a probability value. The "optimize" clause specifies the criteria that need to be optimized; it requires minimization or maximization of some measure.

The query in Fig. 3 retrieves trips/paths from work to home where only the pedestrian and bus modes are allowed. The WHERE clause specifies that the trip must end by 5:00 pm, and that the constraints must be met with probability >0.8. Among all such paths the shortest (by length) is selected.

The query in Fig. 4 asks for the fastest trip from home to work using bus and walking (pedestrian) with a stop at a pharmacy so that the distance from the pharmacy to work is at most 2 km.

The reader is referred to (Booth et al., 2009) where the detailed syntax and semantics of the query language are presented. That work also presents algorithms for processing restricted classes of TranQuyl queries.

2.3. Future work

2.3.1. Data model and semantics

The data model and query language specified in Section 2.2.3 has been proposed primarily for trip planning. However, we need to allow specification of a wide variety of other queries, such as the following ones related to trip execution: retrieving the average speed one mile ahead of the traveler; or retrieving the potential ride-sharing opportunities based on commuters' trips and departure-times; or retrieving all the hazards ahead of a traveler; or retrieving all the vehicles approaching an intersection on the current trip. Note that the last of the above queries may be needed when a car is approaching an intersection that is controlled by a virtual traffic light and all such cars need to coordinate so as to traverse the intersection safely. To permit the above types of queries, the data model and the query language need to be extended. For example, we need to include additional attributes to be associated with the edges and vertices of the network. Note that attribute values of edges may not be stored in a central database but may be computed on demand from the data stored on neighboring vehicles traveling on the network. Such data may be automatically generated by sensors, e.g. speed sensors, but it may also be entered manually (e.g. a voice recording "there is an accident here," or a text message "the escalator in the train station is not working"). In other words, values of attributes of the transportation network may be generated by crowd sourcing of individual travelers.

Finally, we need to allow the specification of uncertainty in the query (in addition to uncertainty in the data) through appropriate constructs, i.e., the specification of fuzzy queries. For example, a commuter may want to query all the near-by taxi cabs; here "near-by" is a fuzzy query construct.

Please cite this article in press as: Wolfson, O., et al. The TranQuyl language for data management in intelligent transportation. Transport. Res. Part C (2012), doi:10.1016/j.trc.2012.02.002

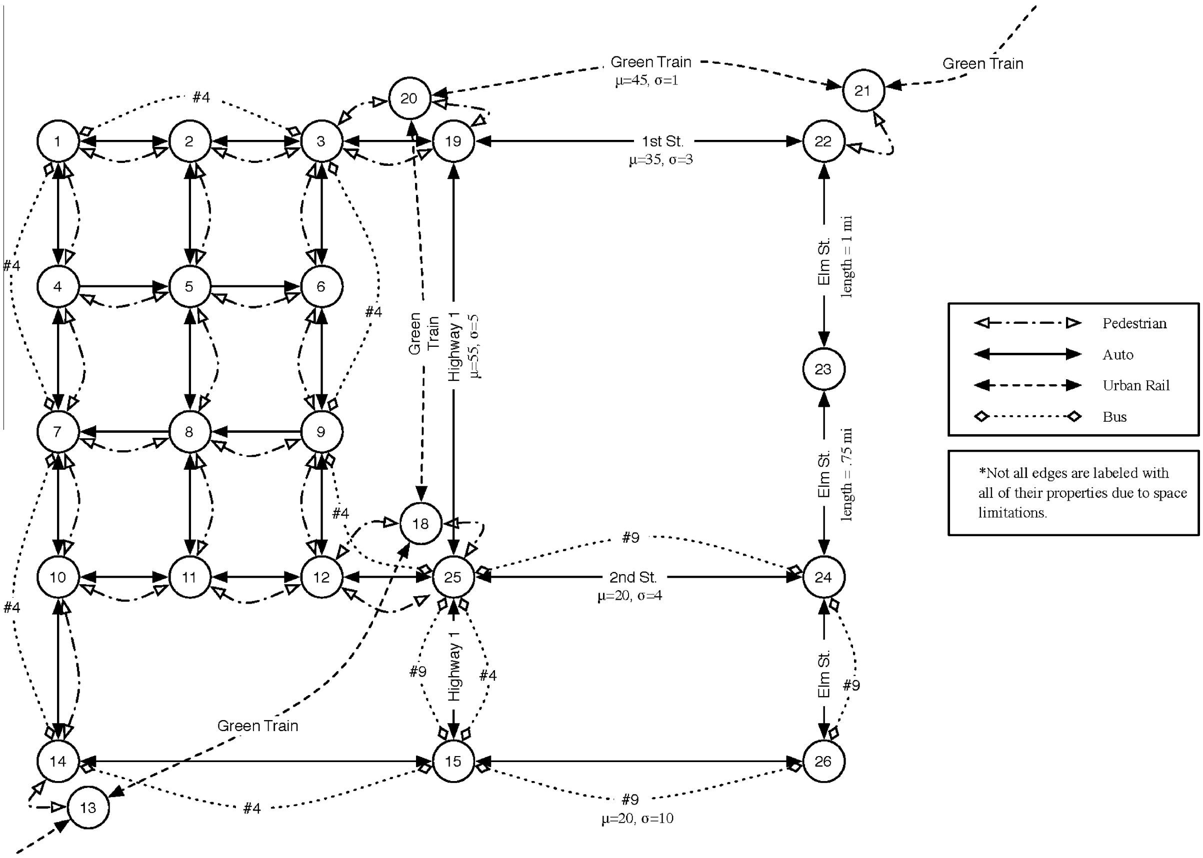

Fig. 1. An example graph of a transportation network.

```
<SELECT *>
<FROM ALL_TRIPS(origin,
destination)>
<WITH STOP_VERTICES>
<WITH MODES>
<WITH CERTAINTY>
<WHERE>
<OPTIMIZE>
```

Fig. 2. Generic query structure.

2.3.2. Graphical query languages

The current version of TranQuyl is textual and has a syntax similar to SQL. Such textual query languages may not be easy to use on the move. We should investigate and develop graphical extensions to TranQuyl. For example, a user should be able to specify the origin, destination together with the stops and other constraints of the required trip on a displayed map. For an example of a graphical routing language see (Biagioni et al., 2009).

2.3.3. Network hierarchies

We will extend the network model to cover indoor spaces for pedestrians. With such an extension a node in the transportation network (e.g., a train station) expands to a separate network for pedestrians. In contrast, observe that the current model, as depicted in Fig. 1, is ''flat''. Another use for hierarchies is in the incorporation of air travel. In this case an air-network node (a city) expands to an urban transportation network. A possible strategy for processing queries on such hierarchies is based on dynamic programming decomposition: generate subqueries on the individual components of the hierarchy and combine the results.

Similarly, the query ''Is the virtual traffic-light at the upcoming intersection green?'' needs a different level of granularity of the transportation network model than an intercity trip planning query. In other words, as with the node expand/collapse discussed above, zoom in/out capabilities on the model are necessary.

```
SELECT *
FROM ALL_TRIPS(work, home)
AS t
WITH MODES pedestrian, bus
WITH CERTAINTY .8
WHERE FINISHES(t) <= 5:00pm
MINIMIZE LENGTH(t)
```

Fig. 3. Query Example 1.

```
SELECT *
FROM ALL_TRIPS(home, work)
AS t
WITH STOP_VERTICES v1
WITH MODES pedestrian, bus
WHERE BEGINS(t) = 8:00AM
  AND "pharmacy" IN v1.facilities
  AND DISTANCE(v1,work) <
2Km
MINIMIZE DURATION(t)
```

Fig. 4. Query Example 2.

3. Query processing in mobile P2P and vehicular networks

3.1. Technical approach

A mobile peer-to-peer (MP2P) network is a set of mobile peers (vehicles, portable devices) that communicate with each other via short-range wireless technologies such as WiFi or DSRC. The MP2P network forms a publish/subscribe system in which the peers communicate reports (publications) and queries (subscriptions) to neighbors directly, and the reports and queries propagate by transitive multi-hop transmissions. Various types of real-time traveler information can be communicated, including safety alerts (e.g., a car in front with a malfunctioning brake light), traffic conditions (possibly represented by multimedia clips), accidents, transit-vehicle on-time performance, parking availability, ride share opportunities, and so on. Existing studies explored two paradigms of query/publish/subscribe in MP2P networks, namely push (publication-to-subscription) and pull (subscription-to-publication). In the push paradigm, reports are proactively disseminated (Cenerario et al., 2011; Gau et al., 2009). In the pull paradigm, queries are proactively disseminated; reports are disseminated as responses to received queries (Delot et al., 2011; Guo et al., 2005; Wu et al., 2007). MP2P networks have also been considered as an augment to the cellular communication (Al-Chikhani et al., 2009; Leung and Chan, 2007). In this case, the report sources reside on the fixed network. Some of the peers download the reports via the cellular communication, and share the reports with the other peers via the short-range communication. We refer to the MP2P networks in which the cellular communication is available as *hybrid MP2P networks*.

Consider hybrid MP2P networks where reports are generated and carried by the peers, e.g., a parking availability report is generated by a peer when the peer leaves a parking slot. In other words, the report sources reside on peers rather than in the fixed network. In a MP2P network, a peer usually does not initially know the network-id's (i.e. cell-phone numbers) of the other peers in the network. However, a peer can communicate directly with other peers within its WiFi transmission range without knowing their network-id.

An environment as described above renders a broad spectrum of possible query processing strategies, along the following design dimensions. First, the peer-to-peer communication may use WiFi or it may use both WiFi and cellular communication (i.e., hybrid). Second, as aforementioned, the query processing may adopt push or pull or the combination of the two. We abstract query processing in a hybrid MP2P network by a paradigm called **WiMaC** (WiFi-communication, Match, Communication). In the WiMac paradigm, since network id's are not known, query processing strategies start with a stage of WiFi dissemination to neighboring peers. The dissemination may be of the query, the reports, or some combination. When a match is found, it may be followed by a second stage of additional cellular or WiFi communication. For example, assume that a query and a report meet at an intermediate peer and they match. Then the report has to be transferred to the query producer by additional communication.

Now the WiMac query processing paradigm can be specialized to particular strategies by considering the possible design choices in terms of push versus pull and utilization of the cellular communication. In preliminary work (Xu et al., 2010) we derived five possible query processing strategies from the WiMac paradigm. We defined the notion of dominance between query processing strategies. Intuitively, strategy A dominates strategy B when each query returns in A a superset of the set of answers it returns in B, each with a response time that is not higher in A than in B; additionally, the communication cost of A is not higher than that of B. Based on this definition we proved analytically that 3 of the 5 WiMac strategies dominate the others. These 3 non-dominated strategies are as follows.

- *Push*: In the first stage of WiMaC reports are disseminated via WiFi. Queries are kept at the producer peer, and a match occurs when a disseminated report arrives at a matching query. There is no second stage.
- *Pull*: in the first stage queries are disseminated via WiFi. When a query reaches the producer peer of a matching report, the producer peer disseminates the report via WiFi to reach the query originator.
- *Push–pull-cellular*: In the first stage reports and queries are disseminated via WiFi. When a report and a matching query meet at an intermediate peer Z, Z sends the report to the query originator via the cellular communication.

3.2. Future work

3.2.1. Query language extensions

TranQuylquery processing and the WiMaC paradigm need to be integrated. Furthermore, TranQuylneeds to be extended with primitives that will provide processing hints. Such primitives will allow, for example: (i) specification of the communication network to be used for query/pub/sub dissemination in cases where more than one network (e.g. WiFi and cellular) is available, and (ii) query-processing bounds on energy consumption for portable devices.

3.2.2. Specialized queries

One specialized query type asks for binary large objects (blobs) such as video/voice clips regarding traffic conditions. This makes query processing even more challenging due to the potential volume of blob data. However, for the purpose of matchmaking, namely finding the reports that satisfy a query, only the metadata descriptions of the reports are needed. For example, the metadata of a multimedia clip may simply include the time and location at which the clip was produced. This characteristic adds another design dimension. That is, due to size-differences, the metadata and blob subreports of a given report may be disseminated independently, and by different means. In prior work (Xu et al., 2010) we incorporated this dimension into WiMaC strategies.

4. Addressing query fuzziness by answer ranking and filtering

Data that is used by ITS applications is usually generated by sensors, whereas travelers pose queries at a higher level of abstraction than that generated by sensors, and mediation is necessary. We will demonstrate the problem using the Emergency Electronic Brake Light (EEBL) application (Section 4.1), and then discuss future work in Section 4.2.

4.1. An example: the EEBL application

4.1.1. The problem

In 2005, the National Highway Traffic Safety Administration (NHTSA) released a document identifying eight potential safety applications which utilize Dedicated Short Range Communication (DSRC) technology (Carter, 2005). The applications were selected based on potential safety benefits they provide. Among them, the EEBL application was determined to be one of three applications to possess high benefit potential. EEBL was defined as an application that alerts drivers of any hard braking done by vehicles in front of them. The idea was to extend drivers' visibility through the emergency brake notifications (see Fig. 5 available on the web). This was described as most helpful in situations where visibility is limited, such as in adverse weather conditions.

According to the NHTSA document, the system design for the EEBL application would work by vehicles automatically disseminating a report each time they perform emergency braking. It was suggested that emergency braking could be defined based on the deceleration rate exceeding a certain predefined threshold. When a report would arrive at a vehicle, the system would check whether the information contained in the report is relevant to the driver. Based on this, an emergency brake warning light would turn on. The determination of how to check whether a report is relevant for the given vehicle was left unspecified, noting only that road lanes could be used as one of the factors. Using the road lane as the only factor, all drivers of the following vehicles in the same lane would see the EEBL warning. Without using other factors, the warning would

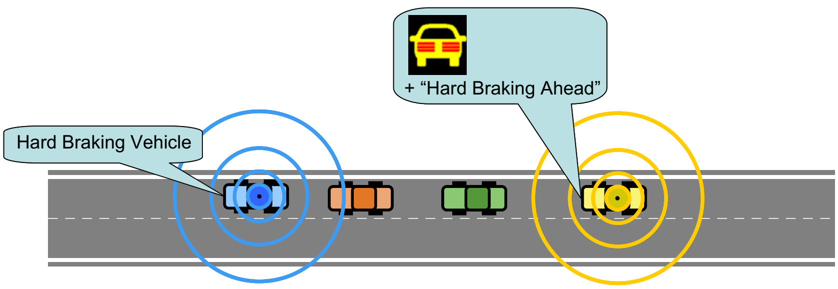

Fig. 5. Illustration of the EEBL application.

hence be seen by all drivers in the lane, with the only restriction being the range of the DSRC device. This would cause many unnecessary EEBL warnings, which might result in drivers becoming desensitized and ignoring the EEBL system. The false warnings could be eliminated by restricting the EEBL warnings to those vehicles for which emergency braking is necessary. This could be done by calculating required deceleration forces given certain report attributes, such as distance, vehicle speeds, or vehicle density. Such approach is commonly done for Cooperative Adaptive Cruise Control applications (Naus et al., 2009). The developed equations however, do not take into account normal driver behavior and might result in drivers to consider the warnings as unnecessary. It may also be difficult to come up with the proper equation that would take into account all the relevant factors. In preliminary work (Szczurek et al., 2011) we compared this analytical approach to the ODaLe principle discussed next.

4.1.2. Possible solution: the observe-driver-and-learn principle

In (Szczurek et al., 2011) we proposed a method for determining whether or not to turn on the EEBL light. It operates on the following Observe-Driver-and-Learn (ODaLe) principle: observe the information (e.g. EEBL reports) that the driver acts upon, and based on it build a statistical machine-learning model. The model is then used to determine the relevance of information arriving in the future. Thus the method operates in two stages:

- a *learning stage* when the driver is "watched" but the EEBL light is not turned on, and
- a *production stage* when the model learned in the first stage is used to determine whether or not to turn on the EEBL light, i.e., in this stage the EEBL light is used.

The first stage uses a machine learning approach for learning reports' relevance. In this method, vehicles check whether emergency braking was done shortly after receiving each report. If so, the report was relevant, otherwise it is not. In other words, emergency breaking is some form of relevance-feedback (a concept used extensively in Information Retrieval). Based on this, training examples are created for a machine learning process which learns a report relevance model. The learned model can then be used to determine the likelihood of an arbitrary report being relevant. The decision to warn the driver is then based on this likelihood. For EEBL the likelihood will depend on several features such as:

- the distance between the reporting and receiving vehicle,
- the density of vehicles on the road (that can be deduced from "here I am messages"), and
- the difference between the velocities of the reporting and receiving vehicles.

The advantage of our proposed method is the ability to combine these individual features into a single score. Also, although the method uses supervised learning, no additional human action is required for learning. Learning is done by simply "watching" (by using the brake sensors) what the driver does anyway. In other words, the method automatically tailors to driver behavior, thus reducing the probability of false warnings during the production stage.

The problem with this approach is that it involves installation of machine learning software in vehicles in order to observe drivers, which is unrealistic. The solution proposed in (Szczurek et al., 2011) is to use a microscopic traffic simulation system such as MITSIM (Yang and Koutsopoulos, 1996). It simulates vehicle movements through a road network, in traffic, using car-following logic and driver/car behavior. We modified MITSIM to enable its usage in ODaLe to train the statistical model. The ODaLe principle feasibility is demonstrated in (Szczurek et al., 2011).

4.2. Future work

4.2.1. Application of the ODaLe principle to general TranQuyl queries

The ODaLe principle can be applied to other safety applications e.g., Blind Spot Warning, Intersection and Curve Approach Warnings, Slippery-Road Warning. The reason is that the EEBL problem epitomizes a more general issue of query fuzziness in Intelligent Transportation. Specifically, the traveler is not able to formulate a precise query, particularly given the fact that answer data is generated by sensors. For example, a driver or traveler is always interested in emergency and safety related information. But this query is fuzzy, and it is hard for the driver to specify exactly which emergency/safety information should be presented to her/him. Too much information may overwhelm, and too little may be dangerous. Emergency information a mile ahead on the route may or may not be relevant depending on factors such as speed, density and visibility. In other words, answers to the safety/emergency query need to be ranked, and presented selectively (e.g. above a cutoff point). The problem is reminiscent of Google's Page Rank, except that in this case the information being ranked and the setup are totally different.

4.2.2. Fine-tune the relevance-estimation

This includes studying how to combine the learning and production stages of ODaLe. This combination is necessary since we expect the ODaLe principle to be applied as follows. A vehicle will be "factory-equipped" with a machine learnt model M, trained based on a simulator such as MITSIM. After the vehicle is purchased, M is continuously modified according to its driver's behavior. And different drivers of the same vehicle will use different models, similarly to the way they use different car-seat adjustments.

Please cite this article in press as: Wolfson, O., et al. The TranQuyl language for data management in intelligent transportation. Transport. Res. Part C (2012), doi:10.1016/j.trc.2012.02.002

4.2.3. ODale as middleware

We should build a general middleware for determining relevance and ranking of low-level sensor answers to fuzzy queries. The middleware should be general-purpose, adaptable, and installable in traffic-simulation systems such as MITSIM. Of course, machine learning for ranking of information is a well studied approach, but it is also general. It needs to be applied specifically to fuzzy queries in TranQuyl.

5. Relevant work

5.1. Transportation information, moving object databases, and graph query languages

The transportation information available to users today generally falls into two categories: form-based and map-based. Many transportation agencies provide web sites that allow users to plan a trip using the public transportation system. They tend to allow the specification of time constraints, mode constraints (some include information for the auto network as well), preferences for walking distance, and how the trip should be optimized (e.g., duration vs. number of transfers). If a valid trip can be constructed the user is presented with an itinerary for its execution. A wide range of algorithms supporting these route-finding queries have been developed to account for the problems with modal transfers, schedules, and cost computation (see e.g., Lozano and Storchi, 2001).

The second common class of planning tools has map-based graphical user interfaces (Google, 2012; VISSIM, 2009; Biagioni et al., 2009). Users may enter their origin and destination via either a form or by clicking points on a map. Unlike most form-based planners, some map-based sites allow for the insertion of multiple stops along the trip and may include some real-time traffic information.

In the area of moving object databases, Güting et al.'s Spatio-Temporal Query Language (STQL) provides an extremely rich data model and query language for modeling moving objects in both open areas (Guting et al., 2000) as well as road networks (Guting et al., 2006).

While STQL enables rich description of some transportation systems, it is not sufficient for our applications. The STQL network model is restricted to only the road network. Multi-modality is a requirement in most urban planning scenarios, and for our purposes must be included in the model at the ground level. There are no explicit types for trip-related concepts. There are no explicit transfers or legs. In order to properly model transportation systems these transportation concepts need to be modeled.

The above references do not address the range of applications discussed here which necessitate the integration of Intelligent Transportation, multimodality, uncertainty, query fuzziness, and their processing in a hybrid P2P, vehicular, and cloud environment.

5.2. Vehicular network query processing

Data dissemination in vehicular networks has been studied before (see e.g., Nadeem et al., 2006; Lee et al., 2009; Cenerario et al., 2011; Delot et al., 2011), but more work is required to integrate the cellular network and the query/response paradigm into the schemes. Grassroots (Goel et al., 2003), SOTIS (Wischoff et al., 2003), and TrafficView (Dashtinezhad et al., 2004) develop environments in which each vehicle contributes a small piece of traffic information to the network based on the P2P paradigm, and each vehicle aggregates pieces of the information into a useful picture of the local traffic information. MobEyes (Lee et al., 2009) is a middleware that is designed for proactive urban monitoring and exploits node mobility to opportunistically diffuse sensed data summaries among neighbor vehicles and to create a low-cost index to query monitoring data. Each of these platforms is designed for a specific application. CarTel (Hull et al., 2006) addresses the application of SQL to an environment in which vehicles transfer collected data to a central database via fixed access points. Some constructs in CarTel's query language, such as "priority" and "deliver order by", are useful for our purpose.

5.3. Answer filtering and relevance of answers

A filtering approach, such as the one proposed in Section 4, is necessary since many TranQuyl queries are fuzzy, such as: retrieve EEBL information from the vehicles ahead; or: retrieve safety related information around me. And the received information in response needs to be ranked, and displayed only if it exceeds a threshold. Now this approach is well studied in Computer Science and Artificial Intelligence. For example, relevance feedback in information retrieval is a form of finding relevance through machine learning. Google's ranking of query results also employs some form of query refinement via machine learning. Now, the application of the principle to each domain is different, and we proposed a way to apply it in Intelligent Transportation.

Acknowledgment

Research supported by NSF DGE-0549489, CCF-0916438, and CNS-1035914.

References

Al-Chikhani, S., Al-Kanj, L., Dawy, Z., 2009. Video Distribution over Wireless Networks with Mobile-to-Mobile Cooperation. Conference on Advances in Computational Tools for Engineering Applications (ACTEA), 2009.

Biagioni, J., Agresta, A., Gerlich, T., Eriksson, J., 2009. Transitgenie: a real-time, context-aware transit navigator (demo abstract). In: The 7th ACM Conference on Embedded Networked Sensor Systems (SenSys 2009), pp. 329–330.

Booth, J., Sistla, P., Wolfson, O., Cruz, I., 2009. A Data Model and Query Language for Urban Transportation Networks. Extending Database Technologies (EDBT), Saint-Petersburg, Russia, March 23–26, 2009, pp. 994–1005.

Carter, A., 2005. The Status of Vehicle-to-vehicle Communication as a Means of Improving Crash Prevention Performance, Tech. Rep. 05-0264, NHTSA. <http://www.nhtsa.gov/DOT/NHTSA/NRD/Multimedia/PDFs/Crash%20Avoidance/2005/CAMP3scr.pdf> (accessed 30.01.12).

Cenerario, N., Delot, T., Ilarri, S., 2011. A content-based dissemination protocol for VANETs: exploiting the encounter probability. IEEE Transactions on Intelligent Transportation Systems 12 (3), 771–782.

Dashtinezhad, S., Nadeem, T., Liao, C., Iftode, L., 2004. A scalable traffic monitoring system. International Conference on Mobile Data Management.

Delot, T., Mitton, M., Ilarri, S., Hien, T., 2011. A routing protocol for query processing in vehicular networks. Mobile Information Systems 7 (4), 329–359.

Gau, V., Huang, C., Hwang, J., 2009. Multimedia broadcasting over dense wireless Ad-Hoc networks. Journal of Communications 4 (9), 614–627.

Goel, S., Imielinski, T., Ozbay, K., Nath, B., 2003. Grassroots: A Scalable and Robust Information Architecture. Technical Report DCS-TR-523, Rutgers University.

Google, 2012. Google Maps. <http://www.maps.google.com/> (accessed 30.01.12).

Guo, M., Ammar, M., Zegura, E., 2005. V3: A vehicle-to-vehicle live video streaming architecture. In: Third IEEE International Conference on Pervasive Computing and Communications, 2005.

Guting, R.H., Bohlen, M.H., Erwig, M., Jensen, C.S., Lorentzos, N.A., Schneider, M., Vazirgiannis, M., 2000. A foundation for representing and querying moving objects. ACM Transactions on Database Systems 25 (1), 1–42.

Guting, R.H., Almeida, T., Ding, Z., 2006. Modeling and querying moving objects in networks. VLDB Journal 15 (2), 165–190.

Hull, B., Bychkovsky, Zhang, Y., Chen, K., Goraczko, M., Miu, A., Shih, E., Balakrishnan, H., Madden, S., 2006. CarTel: a distributed mobile sensor computing system. In: The 4th International Conference on Embedded Networked Sensor Systems, November 2006.

Lee, U., Magistretti, E., Gerla, M., Bellavista, P., Corradi, A., 2009. Dissemination and harvesting of urban data using vehicular sensing platforms. IEEE Transactions on Vehicular Technology (58), 882–901.

Leung, M., Chan, S.H., 2007. Peer-to-peer collaborative video streaming among mobiles. IEEE Transaction on Broadcasting (53), 350–361.

Lozano, A., Storchi, G., 2001. Shortest viable path algorithm in multimodal networks. Transportation Research Part A: Policy and Practice 35, 225–241.

Ma, S., Wolfson, O., Lin, J., 2010. IIP: An event-based platform for ITS applications. In: Proc of the 3rd International Workshop on Computational Transportation Science, San Jose, CA, November 2010.

Nadeem, T., Shankar, P., Iftode, L., 2006. A comparative study of data dissemination models for VANETs. In: The 3rd Annual International Conference on Mobile and Ubiquitous Systems: Networks and Services, 2006.

Naus, G., Vugts, R., Ploeg, J., Molengraft, R.V., Steinbuch, M., 2009. Towards On-The-Road Implementation Of Cooperative Adaptive Cruise Control. ITS World Congress, 2009.

RITA, 2010. Research and Innovative Technology Administration, ITS Strategic Research Plan, 2010-2014, Executive Summary. <http://www.its.dot.gov/strategic_plan2010_2014/index.htm> (accessed 30.01.12).

Szczurek, P., Xu, B., Wolfson, O., Lin, J., 2011. Intelligent transportation systems: when is safety information relevant? IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks (WoWMoM), June 2011, pp. 1–6.

VISSIM, 2009. VISSIM: State-of-the-Art Multi-Modal Simulation. <http://www.ptvag.com/fileadmin/files_ptvag.com/download/traffic/Broschures_Flyer/VISSIM/VISSIM_Brochure_e_2009_HiRes.pdf> (accessed 30.01.12).

Winter, S., Sester, M., Wolfson, O., Geers, G., 2011. Towards a computational transportation science. Journal of Spatial Information Science (2), <http://josis.org/index.php/josis/article/download/39/40> (accessed 30.01.12).

Wischoff, L., Ebner, A., Rohling, H., Lott, M., Halfmann, R., 2003. SOTIS – a self-organizing traffic information system. In: Proceedings of Vehicular Technology Conference (VTC), 2003.

Wu, H., Peng, H., Zhou, Q., Yang, M., Sun, B., Yu, B., 2007. P2P multimedia sharing over MANET. In: Proceedings of the International Conference on Multimedia Modeling, 2007.

Xu, B., Vafaee, F., Wolfson, O., 2009. In-network query processing in mobile P2P databases. In: Proceedings of the ACM SIGPATIAL International Conference on Advances in Geographic Information Systems (ACM GIS), Seattle, November 2009.

Xu, B., Wolfson, O., Lin, J., 2010. Multimedia data in hybrid vehicular networks. In: Proceedings of the International Conference on Advances in Mobile Computing & Multimedia (MOMM), Paris, France, November 8–10, 2010.

Yang, Q., Koutsopoulos, H.N., 1996. A microscopic traffic simulator for evaluation of dynamic traffic management systems. Transportation Research Part C: Emerging Technologies 4 (3), 113–129.

Please cite this article in press as: Wolfson, O., et al. The TranQuyl language for data management in intelligent transportation. Transport. Res. Part C (2012), doi:10.1016/j.trc.2012.02.002