Semantic Traffic-Aware Routing Using the LarKC Platform

The popularity of location-based services and automotive navigation systems calls for a new generation of intelligent solutions to support users in mobility. This article presents a traffic-aware semantic routing service for mobile users based on the Large Knowledge Collider (LarKC) Semantic Web pluggable platform. It proposes a technique for integrating conceptual query answering with statistical learning and operations research algorithms. The presented prototype of a traffic-aware semantic routing service works efficiently with large, heterogeneous information sources and delivers value-added services to mobile users.

Emanuele Della Valle
Politecnico di Milano

Irene Celino and Daniele Dell'Aglio
Cefriel, Politecnico di Milano

Ralph Grothmann, Florian Steinke, and Volker Tresp
Siemens Corporate Research and Technologies

Mobile technology users often turn to location-based services and systems to get directions or information about their surrounding environment. Such technology presents various challenges, and research in different fields provides partial solutions to those needs: operations research solves the routing problem, machine learning addresses traffic forecasting, and semantic technologies manage data integration and information retrieval. Still, the demand for location-based comprehensive solutions overcomes what's currently on offer. Here, we aim to address the comprehensive research challenges inherent to location-based services.

People in (unfamiliar) urban environments might ask questions like, "What museum can I reach in less than 25 minutes if I get into my car at 4 p.m.?" The pieces of information required to answer such questions are usually available, but they're scattered at different locations and aren't interoperable. We aren't aware of any service that can perform the mix of conceptual query answering, machine learning, and operations research required to answer this question.

To address these challenges, we present the Traffic LarKC service, an award-winning prototype based on Semantic Web technologies (the Resource Description Framework [RDF][1] and SPARQL[2]) that seamlessly integrates conceptual query answering, statistical regression, and operations research techniques into a single service fully operational for the city of Milano. We also provide empirical evidence of its efficacy in answering complex semantic queries as well as its efficiency and scalability.

Architecture

To answer the question raised in the introduction, a service must be able to semantically retrieve points of interest (POIs) in a city given a category (conceptual query answering), compute the most suitable path to reach the POI (operations research) by considering traffic conditions and traffic predictions (machine learning), and give a complete answer to the requester in a reasonable time. Our research question was whether we could use Semantic Web technologies to make interoperable the different techniques required to realize such a service. To this end, we needed a platform that would act as a Semantic Web framework, could reuse processing components to leverage different technologies, and would orchestrate the different computations in a single workflow to solve the end user's problem.

In this context, we decided to adopt the Large Knowledge Collider (LarKC),[3] a platform for massive distributed reasoning that removes the scalability barriers of currently existing reasoning systems for the Semantic Web. The LarKC platform has a pluggable architecture to exploit techniques and heuristics from machine learning, operations research, and the Semantic Web. All plug-ins interoperate through a scalable RDF-based data layer. The platform also allows for parallelized and distributed data processing, to improve the resulting applications' scalability.

We selected Milano as the target of our experiments. The datasets our service uses (see Figure 1) are diverse in topic and format. Traffic LarKC retrieves monuments, attractions, exhibitions, and events in Milano from the Linked Open Data cloud,[4] DBpedia,[5] GeoNames (www.geonames.org), LinkedGeoData (http://linkedgeodata.org), and LinkedEvents.[6] Because these data are already in RDF format, consolidated, and interlinked, LarKC can easily retrieve and process them.

Information about Milano's topology and traffic comes from the local mobility agency. The road network shapefile contains approximately 30,000 streets with 15,000 junctions; this dataset semantically describes each street portion with a set of geometrical attributes and

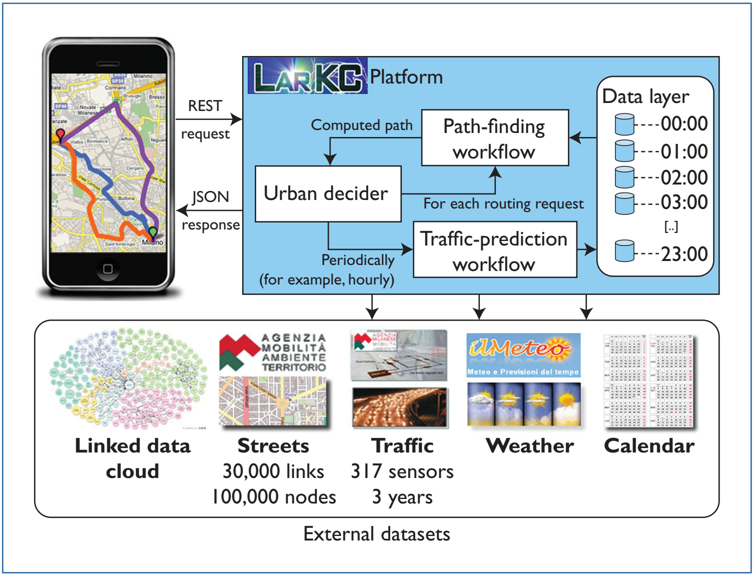


Figure 1. The Traffic LarKC workflows and external datasets. The datasets the service uses are displayed at the bottom; the top-right box represents the workflows the service executes, whereas the top-left part of the figure displays the rendering of the Traffic LarKC response on a mobile device.

flow-related characteristics. Traffic information consists of a three-year time-series record of how many vehicles passed by, sensed every 5 minutes at 317 sensor locations; thus, traffic records add up to more than 10^9 records in a 250-Gbyte database. Additionally, for the same time span, we gathered historical weather data from the Italian website ilMeteo.it (10^8 comma-separated value, or CSV, records) and calendar information (week versus weekend days, holidays, and so on) to consider seasonal effects.

As Figure 1 illustrates, those data are processed by two different LarKC workflows: a traffic prediction workflow that computes traffic forecasts and a path-finding workflow that semantically retrieves POIs and computes the route to reach them.

Traffic LarKC invokes these two workflows at different times: while it re-executes the traffic forecast every hour to update the predictions (batch-time execution), it operates the routing at each end-user request (runtime execution) using the most up-to-date predictions.

Traffic Predictions

Traffic predictions approaches available in literature are based either on simulation or statistical-regression. Simulation-based traffic predictions are easily interpretable (such as DynaMIT; http://mit.edu/its/dynamit.html),

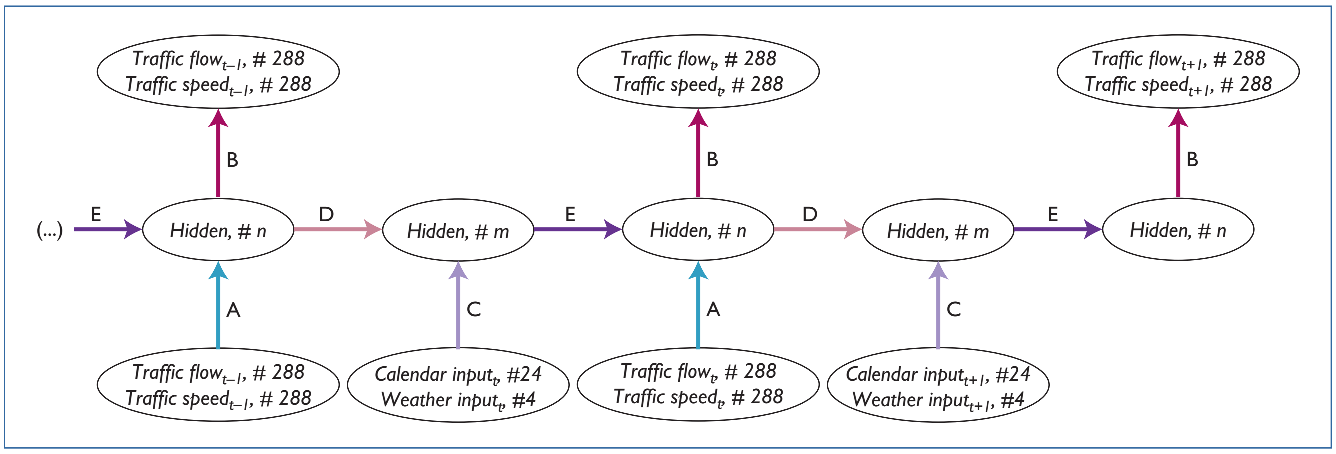

Traffic flow$_{t-1}$, # 288
Traffic speed$_{t-1}$, # 288
Traffic flow$_t$, # 288
Traffic speed$_t$, # 288
Traffic flow$_{t+1}$, # 288
Traffic speed$_{t+1}$, # 288
B
B
B
(...)
E
Hidden, # n
D
Hidden, # m
E
Hidden, # n
D
Hidden, # m
E
Hidden, # n
A
C
A
C
Traffic flow$_{t-1}$, # 288
Traffic speed$_{t-1}$, # 288
Calendar input$_t$, #24
Weather input$_t$, #4
Traffic flow$_t$, # 288
Traffic speed$_t$, # 288
Calendar input$_{t+1}$, #24
Weather input$_{t+1}$, #4

Figure 2. Traffic-prediction recurrent neural network (RNN). The network incorporates bottlenecks to separate time-variant from time-invariant structures.

whereas statistical-regression approaches[7] reduce the need to formulate and integrate detailed assumptions because they're implicit in the data, and the learned system automatically considers them. Our traffic forecasts employ a statistical regression approach by coupling recurrent neural networks (RNNs)[8] with semi-supervised learning (SSL).[9]

Our service invokes the traffic prediction workflow (see Figure 1) recurrently. This workflow performs two steps. First, it uses RNNs to forecast speed and flow at sensor locations for the next four hours, in 5-minute intervals. The predictions are based on traffic observations from the previous 24 hours at the same sensors. After categorizing the predictions into two traffic conditions, normal and congested, the system generalizes the predictions from the sensor locations to all streets of the road network using SSL. It then uses the resulting predictions in conjunction with nominal speed and street lengths to assign estimated travel times. It stores the final results in the data layer as time-stamped "buckets" of information (see Figure 1b). Each bucket is an RDF named graph[10] containing the traffic forecasts in a specific time interval and is annotated with the lower or upper limit of the time validity to enable an efficient retrieval at runtime.

Traffic Predictions at Sensor Locations

The RNN-prediction approach focuses on identifying underlying traffic dynamics. We assume that traffic dynamics are partially driven by an autonomous development (such as the traffic characteristics of different types of days, holidays, and special calendar events) and a variety of external influences (for example, events or weather conditions).

We use an open, discrete-time state-space model, represented by the time-delay RNN structure Figure 2 depicts.[11] The system parameter matrices A through E are identified via *finite unfolding* in time.[4] The RNN architecture includes a coordinate transformation in the form of so-called "bottlenecks," where the neural network uses past flows and speeds both as an input and output. The bottlenecks focus on predicting the traffic dynamics' nonlinear principal components, roughly corresponding to the system's time variants. Because the neural network can reconstruct the traffic dynamics from the variants and invariants, we must forecast only the variants to predict the future time series of the traffic flow and speed.

We performed RNN training for the 317 sensors with training data using Siemens' Software Development Environment for Neural Networks (SENN; www.ct.siemens.com). Within the LarKC plug-in, SENN evaluates the RNNs online with current traffic observations obtained through the data layer.

Network-Wide Generalization

We can interpret the task of deriving traffic predictions for all network streets based on predictions at the sensor locations as the regression problem of mapping locations to traffic conditions. To this end, we employ a Bayesian SSL formulation[9] that exploits the street graph topology to generalize beyond the sensors' locations.

```
1   SELECT DISTINCT ?museum ?path ?length ?travelTime
2   WHERE {
3     ?museum a yago:MuseumsInMilan ;
4             upf:locatedAt ?museumPosition .
5     ?g upf:TrafficRecordsFrom ?from ;
6        upf:TrafficRecordsUntil ?until .
7     FILTER(?from <= "2011-03-01T15:04:01Z"^^xsd:dateTime &&
8            ?until >= "2011-03-01T15:04:01Z"^^xsd:dateTime)
9     GRAPH ?g {
10        ?path a upf:Path;
11              upf:hasStart <START-POS> ;
12              upf:hasGoal ?museumPosition ;
13              upf:hasPathLength ?length ;
14              upf:hasPathTravelTime ?travelTime ;
15              upf:hasPolicy ?policy .
16         ?policy upf:hasMinimizedDimension upf:estimatedTravelTime .
17         FILTER(?travelTime < "PT25M"^^xsd:duration)
18    }
19  }
```

Figure 3. User request internally formalized in SPARQL. This query looks for the museums in Milano and their positions (lines 3 and 4), specifies when the user's journey should take place (lines 5 through 8), and expresses the routing request (lines 9 through 18).

The assumption that neighboring links i have similar traffic conditions f_i is formally encoded in the a priori distribution:

$$p(\mathbf{f}) \propto \exp\left(-\sum_{ij \in E} w_{ij}(f_i - f_j)^2\right),$$

where E is the set of all connected links and w_{ij} is a fixed weight. The RNN traffic predictions at sensors y_i are included via the likelihood

$$p(\mathbf{y}|\mathbf{f}) \propto \prod_i \exp\left(-\frac{1}{\sigma^2}(f_i - y_i)^2\right).$$

The system can then compute the maximum a posteriori traffic estimate for all streets $\mathbf{f}$ via the Bayes rule, which involves solving a large linear system of the size of the number of streets. However, because all involved matrices are sparse, the system can still compute the solution in less than a second, even for the 30,000 streets in our roadmap.

Note that the measured speed and flow values depend strongly on each sensor's field of view, and speed and flow don't necessarily generalize well over the road network. Our solution is to predict only two traffic conditions — that is, normal ($y_i = 0$) and congested traffic ($y_i = 1$). The system classify the speed and flow values into the two traffic conditions by thresholding the traffic speed. For a given street type (small residential street, normal city street, and so on), we then use default values for the two conditions. We discuss the results for both the RNN predictions and the SSL smoothing step in a later section.

Semantic Traffic-Aware Routing

Trip planning considering time constraints is still an open research problem.[12] Consider the user request we presented in the opening: "What Milano museum can I reach in less than 25 minutes if I get into my car at 4 p.m.?" To answer this, we need to query the traffic predictions described in the previous section, together with the semantic descriptions of the city's POIs. In addition, the road network data must be available. Let's look at how our runtime LarKC workflow achieves semantic interoperability between the different modules used to answer the user request.

Querying in SPARQL

Figure 3 shows how the stated user request is internally formalized in SPARQL. The prefix `upf:` refers to our Urban Path-Finding Ontology (http://larkc.cefriel.it/ontologies/urbanpathfinding), which formalizes our semantic model of the traffic- and path-finding domain.

In lines 3–4, this SPARQL query asks for the museums in Milano (here, the instances of the respective Yago concept[13]) and their positions.

To answer this conceptual portion of the query, LarKC employs a reasoner to unfold the Yago category taxonomy and invokes the semantic search engine Sindice[14] for the semantic POI retrieval. Lines 5–8 specify when the journey to the museum should take place. To this end, LarKC selects the named graphs — annotated with the lower/upper bound time limit of validity — that fit the user's needs — that is, that are valid for the requested date and time.

The routing request is expressed in SPARQL at lines 9–18: for the selected graphs (line 9), LarKC must compute a path to the identified museum (lines 10–12) whose duration is compatible with user preferences (line 17). Each path should be described by its length and duration (lines 13–14) and globally minimized via the traffic estimations (line 16). To this end, our Urban Path-Finding Ontology defines three specific routing policies. The first minimizes the path extent (`upf:length`), the second minimizes the duration in absence of traffic (`upf:nominalTravelTime`), and the third minimizes the duration given the traffic forecasts (`upf:estimatedTravelTime`). The actual path computation, according to the policy indicated in the SPARQL query, is delegated to an operations research algorithm.

Efficient Query Evaluation

Our solution optimizes the query evaluation in two ways. First, we decouple the user's routing requests from the traffic-prediction computation. When a user issues a request, traffic forecasts are already available in the LarKC Data Layer as time-stamped named graphs. These graphs are deleted and substituted with new predictions every hour, so forecast computation doesn't affect user request evaluation.

The second optimization considers path finding. We can't answer the full SPARQL query illustrated in Figure 3 using only Semantic Web technologies or only object-spatial databases; Traffic LarKC fragments the full SPARQL query into a conceptual part (POI semantic retrieval) and a routing part and dispatches those subqueries to the different plug-ins registered in the LarKC platform. To compute the paths, the runtime LarKC workflow invokes an operations research plug-in that applies the Dijkstra algorithm. Similarly to D2R,[15] we treat the path computation as a query to a virtual RDF graph, where the Dijkstra-based component is hidden behind a SPARQL-compliant interface. The Traffic LarKC runtime workflow then joins the Dijkstra results with the rest of the data (Milano museums identified by the conceptual part of the query).

Without LarKC, we would have had to build an ad hoc system to put together the different pieces and make them "talk" to each other (for instance, by transforming all data into a GIS-compliant format). LarKC goes well beyond traditional Semantic Web platforms: based on RDF as a means to lightweight data integration, it allows for encapsulating operations research algorithms for path finding within a SPARQL end point.

Evaluation

We evaluated the proposed system with regard to both the quality of the results and the combined system's runtime performance.

Quality

Figure 4 shows the RNN traffic forecasts. In Figure 4a, we show the traffic-flow time series for five exemplary sensors. The RNNs use the past 24 hours of measurements to predict the next four hours. Apart from the predictions' visual plausibility, the RNNs also numerically outperform other standard regression techniques — namely, feed-forward neural networks (MLP) and linear regression, as Figure 4b illustrates. We can see that the RNN's average relative error is significantly lower and its variance much smaller compared to the MLP and linear regression.

Figure 5 shows an example result of the network-wide generalization of sensor traffic predictions. Connected areas of congestion are clearly visible around sensor locations. These results are qualitatively plausible, but their numerical validation isn't currently possible, as no in-between-the-sensors information is available.

The full routing service for the end user is available as a Web application at http://larkc.cefriel.it/traffic-larkc. Figure 6 shows a screenshot of the application.

Currently, the sensor network that collects Milano traffic data isn't designed to deliver information in real time, so we couldn't evaluate our service with real-time data. However, we performed a numerical quality evaluation of our routing service using historical data. We selected 100 random pairs of points and computed

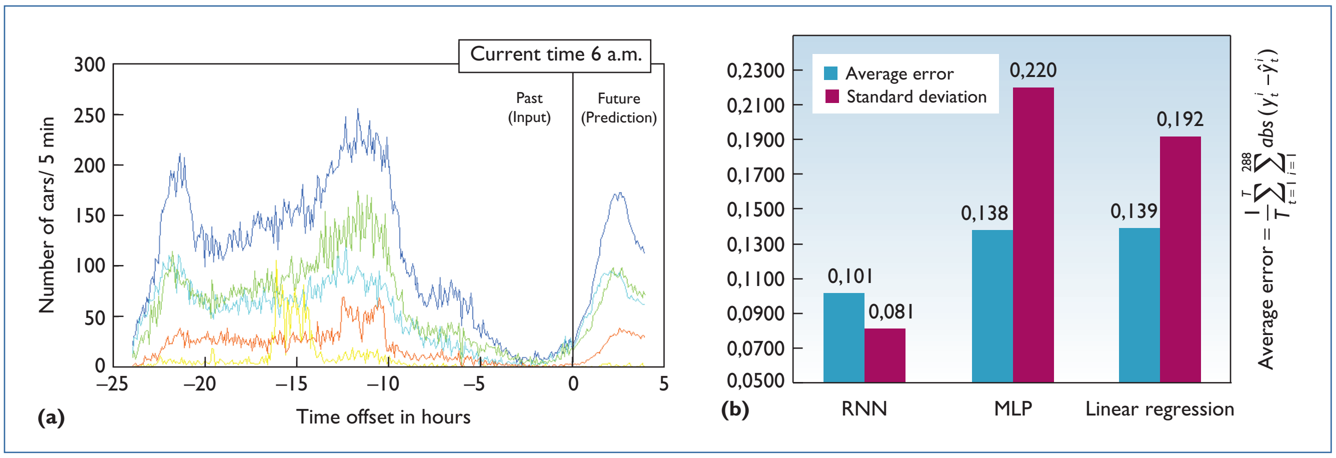


Figure 4. Recurrent neural network (RNN) traffic forecasts. We can see (a) a traffic-flow time series for five exemplary sensors as well as (b) a comparison of the RNN's prediction errors vs. a feed-forward neural network (MLP) and linear regression.

the optimal paths following the three different policies (see Table 1). Compared to the shortest-distance paths, the quickest-path policy with traffic predictions leads to paths that, while 6 percent longer, are 14 percent faster in expected travel time. We confirmed the statistical significance of this advantage using a student's *t*-test with a significance level of 5 percent.

Performance and Scalability

We tested our service's performance and scalability on a six-core AMD Opteron Processor 2431 (2.4-GHz) machine with 8 Gbytes of RAM running Ubuntu 10.04 64-bit. The service activates the traffic-prediction workflow each hour and predicts the traffic for the next four hours on 30,000 streets in 5-minute intervals; thus 600,000 RDF triples are updated in each run based on the newly available predictions. Although the whole run needs 90 seconds, the interaction with the data layer alone — that is, deleting old traffic predictions, reading the necessary data for the RNNs, and rewriting new predictions — takes a share of 81 seconds, leaving only 9 seconds for the actual prediction algorithms. The data-loading cost is clearly an overhead for the LarKC platform. One advantage of the platform was also clear: the RDF representation used in the LarKC architecture permits seamless interoperability between different plug-ins.

For the path-finding workflow, we evaluated runtime performance and scalability via stress tests, issuing concurrent requests across the Internet from an Intel Core 2 (2.16-GHz) machine with 2 Gbytes of RAM and Microsoft Windows XP Professional (SP3). By disabling the LarKC platform's built-in caching feature, the response time is independent of the routing policy, as Figure 7a shows. Thanks to the periodic "batch-time" recomputation of traffic predictions, the time required for runtime routing is independent of the forecast processing. Moreover, by enabling the caching feature, our service replies in a few seconds and shows a sublinear dependency on the number of concurrent requests (see Figure 7b).

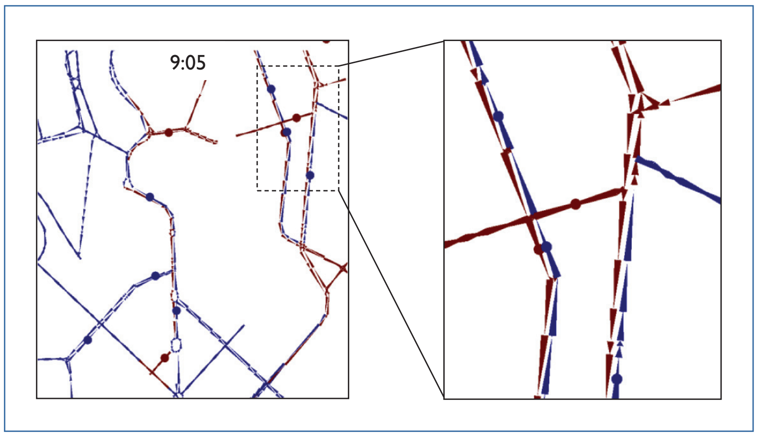
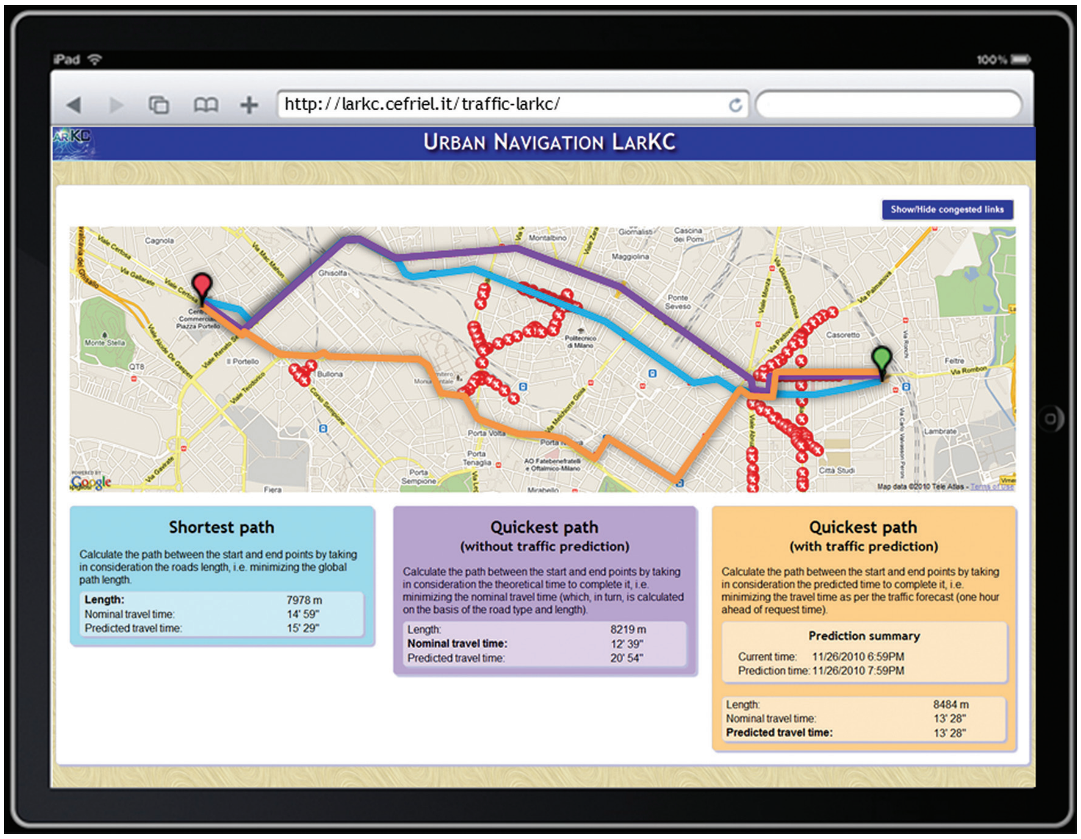


Figure 5. Generalizing traffic conditions at sensor locations (dots) to all links at 9:05 a.m. Blue links have normal traffic conditions, whereas red ones are congested. Different street directions are modeled separately; thus, they can have different traffic conditions.

In the use case we presented, mobile user queries required a service that seamlessly integrates techniques from machine learning (traffic predictions) and operations research


http://larkc.cefriel.it/traffic-larkc/
URBAN NAVIGATION LARKC
Show/Hide congested links

Shortest path

Calculate the path between the start and end points by taking in consideration the roads length, i.e. minimizing the global path length.

Length:	7978 m
Nominal travel time:	14' 59"
Predicted travel time:	15' 29"

Quickest path (without traffic prediction)

Calculate the path between the start and end points by taking in consideration the theoretical time to complete it, i.e. minimizing the nominal travel time (which, in turn, is calculated on the basis of the road type and length).

Length:	8219 m
Nominal travel time:	12' 39"
Predicted travel time:	20' 54"

Quickest path (with traffic prediction)

Calculate the path between the start and end points by taking in consideration the predicted time to complete it, i.e. minimizing the travel time as per the traffic forecast (one hour ahead of request time).

Prediction summary

Current time: 11/26/2010 6:59PM
Prediction time: 11/26/2010 7:59PM

Length:	8484 m
Nominal travel time:	13' 28"
Predicted travel time:	13' 28"

Figure 6. The Traffic LarKC user application as rendered in a Web browser. The paths between the start and end points (green and red pins) are colored differently according to the chosen routing policy.

Table 1. Average relative performance of the different policies for 100 random paths.

Change with regard to shortest-path policy results	Quickest-path policy without traffic prediction (%)	Quickest-path policy with traffic prediction (%)
Length	+3	+6
Nominal travel time (NTT) without traffic prediction	−5	+2
Estimated travel time (ETT) with traffic prediction	+5	−14

(routing) with conceptual query answering (POI semantic retrieval). Here, we demonstrated that a pluggable Semantic Web application framework such as the LarKC platform is suitable for addressing this challenge.

Although we couldn't integrate our prototype with real-time traffic data streams because of the technical limitations of the Milano traffic-management system, such a solution is under discussion for the Milano Expo 2015, and would supply added value to mobile end users.

A general lesson learned is that location information represents a natural basis for integrating several information sources: we were able to sensibly glue together different pieces of data and computational services because they were related to the same physical place. This is one major reason for the recent popularity of mobile location-based services, and also represents a great opportunity for applying semantic technologies.

Acknowledgments
This work was partially supported by the European project LarKC (FP7-215535).

References

1. F. Manola and E. Miller, *RDF Primer*, Word Wide Web Consortium (W3C) recommendation, Feb. 2004; www.w3.org/TR/rdf-primer.

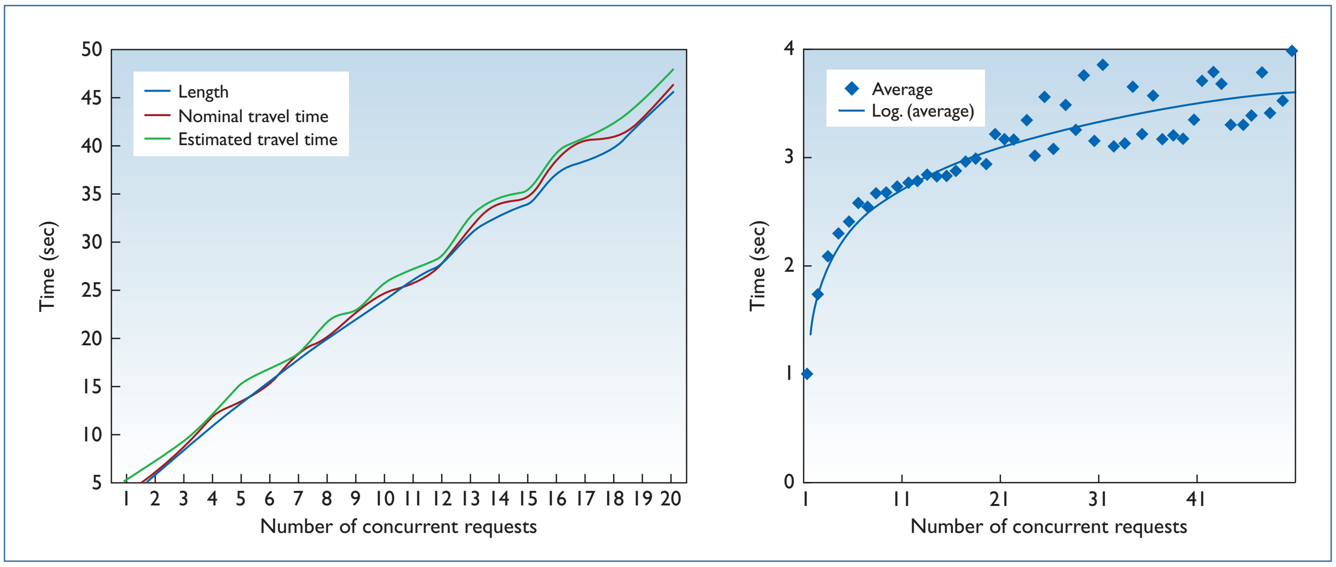


Figure 7. Stress test results on the Traffic LarKC. We can see the results (a) without caching and (b) with caching enabled.

2. E. Prud'hommeaux and A. Seaborne, *SPARQL Query Language for RDF*, Word Wide Web Consortium (W3C) recommendation, Jan. 2008; www.w3.org/TR/rdf-sparql-query.
3. D. Fensel et al., "Towards LarKC: A Platform for Web-Scale Reasoning," *Proc. IEEE Int'l Conf. Semantic Computing* (ICSC 08), IEEE CS Press, 2008, pp. 524–529.
4. C. Bizer, T. Heath, and T. Berners-Lee, "Linked Data — The Story So Far," *Int'l J. Semantic Web and Information Systems*, vol. 5, no. 3, 2009, pp. 1–22.
5. C. Bizer et al., "DBpedia — A Crystallization Point for the Web of Data," *J. Web Semantics*, vol. 7, no. 3, 2009, pp. 154–165.
6. R. Troncy, B. Malocha, and A.T.S. Fialho, "Linking Events with Media," *Proc. I-Semantics 2010*, ACM Press, 2010, pp. 42.1–42.4.
7. S. Clark, "Traffic Prediction Using Multivariate Nonparametric Regression," *J. Transportation Eng.*, vol. 129, no. 2, 2003, pp. 161–168.
8. H.-G. Zimmermann and R. Neuneier, "Neural Network Architectures for the Modeling of Dynamical Systems," *A Field Guide to Dynamical Recurrent Networks*, J.F. Kolen and S. Kremer, eds., IEEE Press, 2001, pp. 311–350.
9. O. Chapelle, B. Schölkopf, and A. Zien, *Semi-Supervised Learning*, MIT Press, 2006.
10. J.J. Carroll et al., "Named Graphs, Provenance, and Trust," *Proc. World Wide Web Conf.* (WWW 05), ACM Press, 2005, pp. 613–622.
11. S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan College Publishing, 1998.
12. W. Souffriau and P. Vansteenwegen, "Tourist Trip Planning Functionalities: State-of-the-Art and Future," *Proc. Int'l Conf. Web Eng.* (ICWE 10), Springer, 2010, pp. 474–485.
13. F.M. Suchanek, G. Kasneci, and G. Weikum, "Yago — A Large Ontology from Wikipedia and WordNet," *J. Web Semantics*, vol. 6, no. 3, 2008, pp. 203–217.
14. E. Oren et al., "Sindice.com: A Document-Oriented Lookup Index for Open Linked Data," *Int'l J. Metadata, Semantics, and Ontologies*, vol. 3, no. 1, 2008, pp. 37–52.
15. C. Bizer and R. Cyganiak, "D2R-Server — Publishing Relational Databases on the Web as SPARQL Endpoints," *Proc. World Wide Web Conf.* (WWW 06), ACM Press, 2006; http://www2006.org/programme/item.php?id=d22.

Irene Celino is a research manager and senior consultant at Cefriel, Politecnico di Milano. Her research interests include the Semantic Web and its application to location-based services, social media analytics, and serious games. Celino has an MSc in biomedical engineering from Politecnico di Milano. Contact her at irene.celino@cefriel.it.

Emanuele Della Valle is an assistant professor of software project management in the Department of Electronics and Information at Politenico di Milano. His research interests include scalable processing of information at the semantic level with a specific focus on streaming and geospatial data. Della Valle has an MSc in computer science from Politenico di Milano. Contact him at emanuele.dellavalle@polimi.it.

Daniele Dell'Aglio is a researcher at Cefriel, Politecnico di Milano. His research interests include the creation and consumption of linked data, mainly in the geographical domain. Dell'Aglio has an MSc in computer science from Politenico di Milano. Contact him at daniele.dellaglio@cefriel.it.

Ralph Grothmann is a senior consultant at Siemens Corporate Research and Technologies. His research interests revolve around modeling dynamical systems with recurrent neural networks in technical and economical applications. Grothmann has a Diploma and PhD in economics from the University of Bremen, Germany. He's a member of the board of the German Operations Research Society. (GOR) Contact him at ralph.grothmann@siemens.com.

Florian Steinke is a research scientist at Siemens Corporate Research and Technologies. His interests range from machine learning and the Semantic Web to the future challenges in generating and distributing energy. Steinke did his doctoral thesis work at the Max-Planck-Institute for Biological Cybernetics. Contact him at florian.steinke@siemens.com.

Volker Tresp is the head of a research team in machine learning at Siemens Corporate Research and Technologies and a professor at the Ludwig Maximilian University of Munich. His research interests include machine learning, data mining, information extraction, and the Semantic Web. Tresp has a PhD from Yale University. Contact him at volker.tresp@siemens.com.