Challenging the Internet of the Future with Urban Computing

Emanuele Della Valle[1,2], Irene Celino[1], Kono Kim[3], Zhisheng Huang[4], Volker Tresp[5], Werner Hauptmann[5], and Yi Huang[5]

[1] CEFRIEL – Politecnico of Milano, Via Fucini 2, 20133 Milano, Italy
`{name.surname}@cefriel.it`
[2] Dip. di Elettronica e Informazione, Politecnico di Milano, Milano, Italy
`emanuele.dellavalle@polimi.it`
[3] Saltlux Inc., Seul, Korea `kono@saltlux.com`
[4] Computer Science Department, Vrije Universiteit Amsterdam, De Boelelaan 1081, Amsterdam, The Netherlands `huang@cs.vu.nl`
[5] Corporate Technology, Siemens AG, Information and Communications, Munich, Germany `{name.surname}@siemens.com`

Abstract. In this paper we present the challenging problem of realizing the Urban Computing vision and in particular we describe the requirements for future mobility management systems. We show that novel multi-disciplinary ideas are required to address the Urban Computing challenge and that only partial solutions can be found today.
The Urban Computing challenge is open and many efforts are needed to address it.

1 Introduction

Our cities must provide answer to very critical questions [6] and among others: "How can we reduce traffic congestion yet stay connected?"

Internet for sure cannot provide an answer on its own, but it is an enabling factor, if not the most important one. A sign that Internet for urban area is growing at a recognizable pace is the rise of the term Urban Computing [1] – the integration of computing, sensing, and actuation technologies into everyday urban settings and lifestyles.

Some years ago, due the lack of data, solving Urban Computing problems looked like a Sci-Fi idea. Nowadays, as demonstrated by the UK government initiative "Show Us a Better Way"[7], a large amount of the required information can be found on the Internet at almost no cost.

For this reason we are challenging the LarKC project[8], which is aiming at a configurable platform for infinitely scalable Semantic Web reasoning [2, 3],

---

[6] `http://www.uli.org/Content/NavigationMenu/AboutULI/CriticalQuestions/About_ULI_Critical_Q.htm`
[7] `http://www.showusabetterway.co.uk/`
[8] `http://www.larkc.eu`

to support the realization of an innovative solution to traffic management. We have been working in this area for years and we can derive from our previous experiences challenging requirements not only for the LarKC project, but also for the entire community working on complex relationship of the Internet with space, places, people and content.

In the rest of the paper, we identify the problem we want to untangle (Section 2) from which we derive requirements for Urban Computing (Section 3). In Section 4, we provide a short description of the partial solutions we are working on, whereas, in the concluding Section 5, we briefly discuss the potential impact of Urban Computing.

2   Sustainable Mobility

Mobility demand has been growing steadily for decades and this growth is foreseen to continue in the future. For many years, the primary way of dealing with this increasing demand has been the increase of the roadway network capacity, by building new roads or adding new lanes to existing ones. However, financial and ecological considerations are posing increasingly severe constraints on this process. Hence, there is a need for additional intelligent approaches designed to meet the demand while more efficiently utilizing the existing infrastructure and resources.

2.1   A Challenging Use Case

This use case shows the added value of (1) collecting a broad set of information about mobility, (2) integrating it and (3) using it to support a citizen that has to go to Milan from Varese (another city in Lombardy region).

- Actors:
    - Carlo: a citizen living in Varese (60 KM North-West of Milano).
    - MUCS: the fictitious Milan Urban Computing System.
- Story Board:
    1. Carlo arranged a meeting in Milan city center for the day after at 11.00.
    2. Willing to plan the travel, he accesses MUCS .
    3. Carlo fills in the required data:
        - FROM: Varese tomorrow after 8.00 ;
        - TO: Milan city center before 11.00;
        - USING: my private car.
    4. MUCS works as Google Maps does today and gives the resulting driving directions, but, instead of saying that such a travel requires 50 minutes, MUCS explains Carlo that he should leave home after 9.00 (when the usual commuters traffic on the A8 motorway is almost over).
    5. MUCS also asks Carlo if he wants to be informed via SMS about traffic conditions and possible alternatives.
    6. Carlo agrees and exits MUCS.

7. The day after an accident involving multiple vehicles take place at 8.15 on the A8 motorway.
8. MUCS estimates that an accident of such kind will result in a congestion of A8 until 10.00, therefore it checks if any planned travel is at risk. It finds Carlo's travel.
9. MUCS checks if Carlo can take an alternative drive, but no alternatives are found to allow Carlo get to Milan in time for his meeting.
10. MUCS checks if Carlo can take public transportation instead. It founds two alternatives:
   (a) Railroad "LeNord" and Subway M3:
      - 8.39 Varese Casbeno - 10.03 Milano Repubblica;
      - take M3 from Repubblica[9] to Duomo (average waiting time 7 minutes, average duration 5 minutes);
   (b) Railroad "FS" and Subways M2 and M3:
      - 8:43 Varese Stato - 9.55 Milano Garibaldi;
      - take M2 from Garibaldi to Centrale (average waiting time 3 minutes, average duration 7 minutes);
      - take M3 from Centrale to Duomo (average waiting time 7 minutes, average duration 8 minutes).
11. MUCS sends an SMS to Carlo informing him that a accident is holding up A8 and he'd better use public transportations; two itineraries have been already prepared for him.
12. Carlo accesses the MUCS service and checks the alternatives. He chooses the first one and uses the ticket-less option to buy the train ticket.

2.2   Challenging Problems

Public authorities have taken steps in the direction to support this use case, but very complex problems has to be solved. Control centers for mobility management have to be connected to different devices (such as detectors on roads, cameras, traffic lights, etc.) and require sophisticated tools for traffic modeling, estimation, prognosis and decision support.

Traffic System Infrastructure Setup Today, in a typical information infrastructure for real-time traffic control that can be found in different cities usually the following basic components can be discriminated. There are sensors (e.g. loop detectors, cameras, traffic eyes, radar detectors) on major roads recording several traffic magnitudes such as vehicle speed (km/h), traffic flow (vehicle/h) and occupancy or traffic density, i.e., the percentage of time the sensor is occupied by a vehicle (vehicles/km). The distance between successive sensors on a freeway is typically in the order of about 500 meters. The information is periodically transmitted to a control center. The control center also receives information about the current state of control devices. Such control devices include traffic signals at

---

[9] Repubblica is both the name of the train station and of the subway station, but they are two different places.

intersections, traffic signals at sideways entry-ramps, variable message signs that can display different messages to drivers (e.g., warning about existing congestions, accidents or alternative path recommendations), radio advisory systems to broadcast messages to drivers, and reversible lanes (i.e., freeway lanes whose direction can be selected according to the current and expected traffic demand). In the control center, operators interpret the sensor data and detect the presence of problems and their possible causes. Problems are congested areas at certain locations caused by lack of capacity due to accidents, excess of demand, like rush hours, etc. In addition, operators determine control actions to solve or reduce the severity of existing problems. For instance, they can recommend to increase the duration of a phase (e.g. green time at a traffic signal) or they may suggest displaying certain messages on some variable message signs to divert traffic.

Recent developments not only consider stationary traffic data provided by standard detectors, but also allow to integrate floating car data (FCD), and an increasing number of operators of advanced traffic management systems also use mobile traffic data.

Traffic Modeling and Estimation An analysis of the current and predicted traffic state in the entire road network and the identification of reserve capacities comprise the basis for advanced city traffic management and navigation solutions. Mobile and stationary sensors collect the appropriate traffic data and transmit it to a central unit. Similarly to the weather forecast, the different and heterogeneous information sources are combined to obtain an estimation of the traffic state during a period ranging from minutes to hours or even longer. Thus, a comprehensive knowledge base is built up to support optimal individual route guidance.

Innovative technologies are required in order to process and integrate the resulting collection of distributed information bits within a complex, diverse information environment. Here, a major task is the provision of appropriate solutions for the integration and fusion of heterogeneous information sources, where each source of information can have distinct characteristics with respect to availability, precision, reliability, resolution and representation.

Reacting to a Changing Environment However, as the use case above shows, deploying an infrastructure, modeling and estimating traffic alone is not sufficient; reacting to changes and suggesting other possible solutions is also important. Traffic is just one aspect of mobility. Private cars are just one of the possible means of transportations. Sometimes public transportation can be by far the best choice.

In the storyboard, Carlo is proposed by MUCS with an alternative solution that depends on the ability of MUCS to collect on-the-fly information about all means of transportation, estimating (based on historical data) that the resolution of the accident will take longer because it took place in the rush hours, comparing a solution using private car with others that use public transportations and proposing Carlo valid alternatives.

3 Requirements

In this section, we will investigate requirements of Urban Computing. As argued before, we are particularly interested in the reasoning requirements for LarKC, but we believe such requirements interesting for the entire community working on the complex relationship of the Internet with space, places, people and content.

3.1 Coping with Heterogeneity

Dealing with heterogeneous data has been appealed for long time in many areas in computer science and engineering, which include database systems, multi-media application, network systems, and artificial intelligence. Here, we would like to propose a comprehensive notion of heterogeneity processing for semantic technologies. We distinguish the following different levels of heterogeneity: Representative Heterogeneity, Semantic Heterogeneity, and Default Heterogeneity.

Representative Heterogeneity means semantic data are represented by using different specification languages. Systems supporting Representative Heterogeneity would allow for semantic data specified by multiple semantic languages, rather than using a single metadata or ontology language, like OWL or RDF/RDFS. However, note that different representation of semantic data does not necessarily mean that they have different semantics. That would be different from Semantic Heterogeneity discussed in the following.

Urban Computing-related data can come from different and independent data sources, which can be developed with traditional technologies and modeling methods (e.g., relational DBMS) or expressed with "semantic" formats and languages (e.g., RDF/S, OWL, WSML); for example, geographic data are usually expressed in some geographic standard[10], events details are published on the Web in a variety of forms, traffic data are stored in databases; etc. The integration and reuse of those data, therefore, need a process of conversion/translation for the data to become useful together.

Semantic Heterogeneity means the systems allow for multiple paradigms of reasoners. For instance, many applications of Urban Computing may need different reasoners for temporal reasoning, spatial reasoning, and causal reasoning. However, it does not necessarily mean that we have to develop a single but powerful reasoner which can cover all of those reasoning tasks. A system which supports Semantic Heterogeneity would find a way to allow multiple single-paradigm-based reasoners to achieve the result of Semantic Heterogeneity.

Some data related to Urban Computing need precise and consistent inference; e.g., knowing if two roads are connected for a given kind of vehicle; telling that at a given junction all vehicles, but public transportation ones, must go straight; checking if private cars are allowed to enter a specific urban area. Other data need approximate reasoning or imperfect estimations; e.g., calculating the probability of a traffic jam given the current traffic conditions and the past history.

---

[10] http://en.wikipedia.org/wiki/Geographic_Data_Files

Therefore, the requirement is for different kinds of techniques and reasoners to deal with those kinds of data; moreover, another requirement is for a system which dynamically selects and runs a specific reasoner on the basis of the available data and the desired processing tasks.

By **Default Heterogeneity**, we mean that systems support for various specification defaults of semantic data. Well-known specification defaults of semantic data are closed world assumption, open world assumption, unique name assumption and non-unique name assumption. In the Semantic Web community, it is widely accepted that semantic data for the Web should take the open world assumption and the non-unique name assumption, as taken by the popular ontology language OWL.

However, as we have observed in many applications of Urban Computing, we should not commit to any single specification default. Take the example of traffic and transportation ontologies: although in many cases we can take the open world assumption and non-unique name assumption, because of our limited knowledge and information about the data, sometimes it is much convenient to take a *local* closed world assumption. For example, for a time table of a bus station, it is well reasonable to assume that the information about the bus schedule in the time table is locally complete, in the sense that if you cannot find any information about a bus which is scheduled at specific time, it would mean that there are no bus scheduled for that time. The same scenario is also applied to a city map: if there is no information which states a road connects two streets directly on the map, that would mean that there is no road which connects those two streets directly.

The same applies to Unique Name Assumption. Consider the use case in Section 2 and in particular the fact that Repubblica is both the name of the train station and of the subway station, but they are two different places. If MUCS has to calculate a trip and Carlo is aware that MUCS will use multiple means of transportation then MUCS can ignore that the two Repubblica stations are not exactly the same one. If, on the contrary, Carlo wanted only to use subways, then MUCS cannot assume that the two Repubblica station are one physical place.

The examples above show that the semantic systems of Urban Computing should support multiple specification defaults. It should allow users or knowledge engineers feel free to state any data with any reasoning assumption. Some part of semantic data may be based on the open world assumption, and some part may be based on the closed world assumption.

3.2   Coping with time-dependency

Knowledge and data can change over the time. For instance, in Urban Computing names of streets, landmarks, kind of events, etc. change very slowly, whereas the number of cars that go through a traffic detector in five minutes changes very fast. This means that the system must have the notion of "observation period", defined as the period when we the system is subject to querying.

Moreover the system, within a given observation period, must consider the following four different type of knowledge and data:

– invariable knowledge:
  – it includes obvious terminological knowledge (such as an address is made up by a street name, a civic number, a city name and a ZIP code) and
  – less obvious nomological knowledge that describes how the world is expected to be (e.g., given traffic lights are switched off or certain streets are closed during the night) or to evolve (e.g., traffic jams appears more often when it rains or when important sport events take place).
– Invariable data: they not change in the observation period, e.g. the names and lengths of the roads.
– Periodically changing data: they change according to a temporal law that can be
  – Pure periodic law, e.g. the fact that every night at 10pm Milan west-side overpass road closes; or
  – Probabilistic law, e.g. the fact that a traffic jam is present in the west side of Milan due to bad weather or due to a soccer match is taking place in San Siro stadium.
– Event driven changing data: they are updated as a consequence of some external event and they can be further characterized by the mean time between changes:
  – Fast, as an example consider the intensity of traffic (as monitored by sensors) for each street in a city;
  – Medium, as an example consider roads closed for accidents or congestion due to traffic;
  – Slow, as an example consider roads closed for scheduled works.

Periodically changing data and event driven changing data are best represented as data streams, unbounded sequences of time-varying data elements. Data streams occur in a variety of modern applications, such as network monitoring, traffic engineering, sensor networks, RFID tags applications, telephone call records, financial applications, Web logs, click-streams, etc. The very nature of Traffic Management can be explained by means of data streams, representing real objects that are monitored at given locations: cars, trains, crowds, ambulances, parking spaces, and so on.

3.3   Coping with Noisy, Uncertain and Inconsistent Data

We distinguish the following different levels of data uncertainty and inconsistency.

– Noisy Data: part of data are completely useless or semantically meaningless.
– Inconsistent Data: parts of data are in logical contradiction with each another, or are semantically impossible.
– Uncertain data: the semantics of data are partial, incomplete, or they are conceptually arranged into a range with multiple possibilities.

Traffic data are a very good example of such data. Different sensors observing the same road area give apparently inconsistent information. For example, a traffic camera may say that the road is empty whereas an inductive loop traffic detector may tell 100 vehicles went over it. The two information may be coherent if one consider that a traffic camera transmits an image per second with a delay of 15-30 seconds, whereas an inductive loop traffic detector tells you the number of vehicles that when over it in 5 minutes and the information may arrive to you 5-10 minutes later.

Moreover, a single data coming from a sensor in a given moment may have no certain meaning. For example, consider an inductive loop traffic detector, it it tells you 0 car went over, what does it mean? Is the road empty? Is the traffic completely stuck? Did somebody park the car above the sensor? Is the sensor broken? Combining multiple information from multiple sensors in a given time window can be the only reasonable way to reduce the uncertainty.

4   Partial Solutions

This work is part of the on-going research project LarKC which aims at building very large-scale manipulation of information ("semantic computing at Web scale"). We are envisioning a set of partial solutions to address the challenges of Urban Computing including: Traffic Prediction using recurrent neural networks, Data Scheduling to address scalability and Stream Reasoning to address time-dependency.

4.1   Predicting Traffic Using Recurrent Neural Networks

Given that a forecast model should focus on the underlying dynamics of the traffic flow and external influences on the traffic volume should be incorporated in the model, we intend to use time-delay recurrent neural networks for the traffic predictions [4]. With this approach we presume that the traffic volume is the outcome of an open dynamical system which combines an autonomous development with external influences (e.g. calendar effects, special events etc.).

Recurrent neural networks offer a new way to model (nonlinear, high dimensional) open dynamical systems based on time series data. Our recurrent neural networks are formulated as state space models in discrete time to identify the traffic dynamics and the impact of the external influences [5].

In state space formulation a recurrent neural network is described by a hidden state-transition- and an output-equation. The temporal equations are transformed into a spatial neural network architecture using shared weights (so-called unfolding in time) [6].

Prior knowledge about the application (e.g. topology of the traffic network or the temporal structure of the traffic flows) can be easily incorporated in the neural network architecture. For instance, an error correction mechanism can be used to consider the impact of unplanned construction sites, traffic accidents or holdups. This is also the key for robust forecasting [7].

4.2   Data Scheduling

The idea of data scheduling takes inspiration from memory management techniques developed and adopted in computer systems and software engineering (e.g., garbage collection, memory caching and direct memory access).

Large scale data are organized at different memory levels based on their relevance and on the context of applications: working data, which should be accessed by systems immediately without any over-heading cost; neighboring data, which can be accessed by the system with a moderate cost; and remote data, which can be accessed by the system with a significant amount of cost.

The research problem is finding automatic ways to move data from higher access cost memory into lower access cost memory and vice versa. Such memory shift should take place in parallel with reasoning.

4.3   Stream Reasoning

Periodically changing data and event driven changing data are best represented as data streams. Processing of data streams has been largely investigated in the last decade [8] and specialized systems have been developed. While reasoners are year after year scaling up in the classical, time invariant domain of ontological knowledge, reasoning upon rapidly changing information has been neglected or forgotten. By coupling reasoners with powerful, reactive, throughput-efficient stream management systems, we introduce the concept of Stream Reasoning [9]. We expect future realization of such a concept to have a strong impact on Urban Computing because it enables reasoning in real time, at a throughput and with a reactivity not obtained in previous works.

5   Conclusions

In this paper we focus on presenting the Urban Computing challenge and in particular some requirements for future mobility management systems. We also presented some novel multi-disciplinary ideas about ways to address the Urban Computing challenge by partially satisfying one or more requirements. More solutions and, in particular, broader ones should be explored. As a matter of fact, if we were able to cope with requirements present in Section 3 we would be able to solve a broad range of Urban Computing problems. Such problems include:

– City Planning: Urban Computing applications can extract statistics and synthetic descriptions of citizens' movements, habits and opinions in order to position new housing complex, office buildings, shops, parking lots, green areas and to optimize public and private transportation routes and timetables. The City Planning can also lower pollution and enhance energy savings.
– Tourism and Culture: Urban Computing applications analyze tourists' movements and enhance the appeal of current places of interest and create targeted promotional campaigns to increase tourism.

– Public Safety: Urban Computing applications can perform continuous statistical analysis of people movements to find abnormal behavior and correlate them with the ones coming from law enforcement and public protection forces to enhance city safety.

Acknowledgments

References

1. Kindberg, T., Chalmers, M., Paulos, E.: Guest editors' introduction: Urban computing. IEEE Pervasive Computing **6**(3) (2007) 18–20
2. Fensel, D., van Harmelen, F., Andersson, B., Brennan, P., Cunningham, H., Della Valle, E., Fischer, F., Huang, Z., Kiryakov, A., il Lee, T.K., School, L., Tresp, V., Wesner, S., Witbrock, M., Zhong, N.: Towards larkc: a platform for web-scale reasoning, IEEE International Conference on Semantic Computing (ICSC 2008) (2008)
3. Fensel, D., van Harmelen, F.: Unifying reasoning and search to web scale. IEEE Internet Computing **11**(2) (2007)
4. Zimmermann, H.G., Neuneier, R.: Modeling dynamical systems by recurrent neural networks. In Ebecken, N., Brebbia, C., eds.: Data Mining II, WIT Press (2000) 557–566
5. Zimmermann, H.G., Neuneier, R.: Neural network architectures for the modeling of dynamical systems. In Kolen, J., Kremer, S., eds.: A Field Guide to Dynamical Recurrent Networks, IEEE Press (2001) 311–350
6. Haykin, S.: Neural Networks. A Comprehensive Foundation. Macmillan College Publishing, New York (1994)
7. Zimmermann, H.G., Neuneier, R., Grothmann, R.: Modeling of dynamical systems by error correction neural networks. In Soofi, A., Cao, L., eds.: Modeling and Forecasting Financial Data, Techniques of Nonlinear Dynamics, Kluwer Academic Publishers (2002) 237–263
8. Garofalakis, M., Gehrke, J., Rastogi, R.: Data Stream Management: Processing High-Speed Data Streams (Data-Centric Systems and Applications). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2007)
9. Della Valle, E., Ceri, S., Barbieri, D.F., Braga, D., Campi, A.: A first step towards stream reasoning. In: Proceedings of the Future Internet Symposium. (2008)