Smart Traffic Analytics in the Semantic Web with STAR-CITY: Scenarios, System and Lessons Learned in Dublin City

Freddy Lécué, Simone Tallevi-Diotallevi, Jer Hayes,
Robert Tucker,Veli Bicer, Marco Sbodio, Pierpaolo Tommasi

IBM Research, Dublin Research Center
Damastown Industrial Estate, Dublin, Ireland
{*(firstname.lastname)@ie.ibm.com*}

Abstract

This paper gives a high-level presentation of STAR-CITY, a system supporting semantic traffic analytics and reasoning for city. STAR-CITY, which integrates (human and machine-based) sensor data using variety of formats, velocities and volumes, has been designed to provide insight on historical and real-time traffic conditions, all supporting efficient urban planning. Our system demonstrates how the severity of road traffic congestion can be smoothly analyzed, diagnosed, explored and predicted using semantic web technologies. Our prototype of semantics-aware traffic analytics and reasoning, illustrated and experimented in Dublin Ireland, but also tested in Bologna Italy, Miami USA and Rio Brazil works and scales efficiently with real, historical together with live and heterogeneous stream data. This paper highlights the lessons learned from deploying and using a system in Dublin City based on Semantic Web technologies.

Keywords: Semantic web, Reasoning system, Intelligent system, Traffic analytics, Smart traffic

1. Introduction and Related Work

As the number of vehicles on the road steadily increases and the expansion of roadways remains static, congestion in cities became one of the major transportation issues in most industrial countries [1]. Urban traffic costs 5.5 billion hours of travel delay and 2.9 billion gallons of wasted fuel in the USA only, all at the price of $121 billion [2]. Even worse, the costs of extra time and wasted fuel has quintupled over the past 30 years. It also used to (i) stress and frustrate motorists, encouraging road rage and reducing health of motorists [3], and (ii) interfere with the passage of emergency vehicles traveling to destinations where they are urgently needed. All are examples of negative effects of congestion in cities.

Three ways can be considered to reduce congestion [4]; one is to improve the infrastructure e.g., by increasing the road capacity, but this requires enormous expenditure which is often not viable in many urban areas. Promoting public transport in large cities is another way but it is not always convenient. Another solution is to determine where, when, and why congestion will be occurring, which will support transportation departments and their managers to proactively manage the traffic before congestion is reached e.g., changing traffic light strategy or re-routing for efficient urban planning.

STAR-CITY[1,2] (**S**emantic **T**raffic **A**nalytics and **R**easoning for **CITY**), as a system which integrates heterogeneous data in terms of format variety (structured and unstructured data), velocity (static and stream data) and volume (large amount of historical data), has been mainly designed to provide such insights on historical and real-time traffic conditions.

[1] Video (.avi, .mov, m4v format) available: http://goo.gl/TuwNyL

[2] Dublin Live system: http://dublinked.ie/sandbox/star-city/

Type	Sensing	Data Source	Description	Format	Temporal Frequency (s)	Size per day (GBytes)	Data Provider (all open data)
Stream Data	Static	Journey times across Dublin City (47 routes)	Dublin Traffic Department's TRIPS system[a]	CSV	60	0.1	Dublin City Council via dublinked.ie[b]
		Road Weather Condition (11 stations)		CSV	600	0.1	NRA[c]
		Real-time Weather Information (19 stations)		CSV	[5, 600] (depending on stations)	[0.050, 1.5] (depending on stations)	Wunderground[d]
	Dynamic	Dublin Bus Stream	Vehicle activity (GPS location, line number, delay, stop flag)	SIRI: XML-based[e]	20	4-6	Dublin City Council via dublinked.ie[f]
		Social-Media Related Feeds	Reputable sources of road traffic conditions in Dublin City	Tweets	600	0.001 (approx. 150 tweets per day)	LiveDrive[g] Aaroadwatch[h] GardaTraffic[i]
Quasi Stream	Dynamic	Road Works and Maintenance		PDF	Updated once a week	0.001	Dublin City Council[j]
		Events in Dublin City	Planned events with small attendance	XML	Updated once a day	0.001	Eventbrite[k]
			Planned events with large attendance			0.05	Eventful[l]
Static	Static	Dublin City Map (listing of type, junctions, GPS coordinate)		ESRI SHAPE	No	0.1	Open StreetMap[m]

[a] Travel-time Reporting Integrated Performance System - http://www.advantechdesign.com.au/trips
[b] http://dublinked.ie/datastore/datasets/dataset-215.php
[c] NRA - National Roads Authority - http://www.nratraffic.ie/weather
[d] http://www.wunderground.com/weather/api/
[e] Service Interface for Real Time Information - http://siri.org.uk
[f] http://dublinked.com/datastore/datasets/dataset-289.php
[g] https://twitter.com/LiveDrive
[h] https://twitter.com/aaroadwatch
[i] https://twitter.com/GardaTraffic
[j] http://www.dublincity.ie/RoadsandTraffic/ScheduledDisruptions/Documents/TrafficNews.pdf
[k] https://www.eventbrite.com/api
[l] http://api.eventful.com
[m] http://download.geofabrik.de/europe/ireland-and-northern-ireland.html

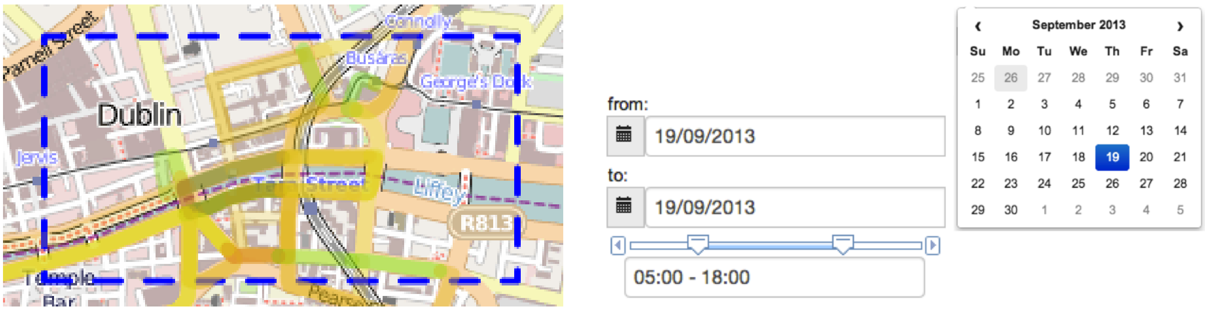
Table 1: (Raw) Data Sources for Dublin City Traffic Scenario

Most of the existing modern traffic systems such as US TrafficView[3] [5], Italian 5T[4] mainly focus on monitoring traffic status in cities using dedicated sensors (e.g., loop indiction detectors), all exposing numerical data. Others, more citizen-centric such as the traffic layer of Google Maps provide real-time traffic conditions and estimation but do not deliver insight to interpret historical and real-time traffic conditions. Basic in-depth but semantics-less state-of-the-art analytics are employed, limiting also large scale real-time data integration. Therefore, context-aware computing together with reusability of the underlying data is quite limited.

On the contrary STAR-CITY strongly relies on interpreting the semantics of contextual information for deriving innovative insights i.e., analysis, diagnosis [6, 7], contextual exploration [8], and more accurate traffic condition forecasting using recent research work in semantic predictive reasoning [9]. Table 1 reports all data sources processed by STAR-CITY in the Dublin scenario with respect to their velocity i.e., static, quasi stream, stream. They report various types of information coming from static or dynamic sensors, exposed as open, public data and described along heterogeneous formats. All rows in grey are data sets not used for traffic diagnosis [6], but required for prediction.

The novelty of STAR-CITY lies in the ability of the system to ingest highly heterogeneous real-time data and perform various types of inferences i.e., analysis, diagnosis, exploration and prediction. These inferences are all elaborated through a combination of various types of reasoning i.e., (i) Description Logic (DL) $\mathcal{EL}^{++}$-based i.e., distributed ontology classification-based subsumption [10], (ii) rules-based i.e., pattern association [9], (iii) machine learning-based i.e., entities search [8] and (iv) stream-based i.e., correlation [9], inconsistency checking [7]. STAR-CITY completely relies on the W3C semantic web stack for representing semantics of information and delivering inference outcomes. Currently experimented in Dublin Ireland,

[3] https://trafficview.org/
[4] http://www.5t.torino.it/5t/



from: 19/09/2013

to: 19/09/2013

05:00 - 18:00

‹ September 2013 ›

Su	Mo	Tu	We	Th	Fr	Sa
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

(a) Spatial (Map Area) Selection

(b) Temporal (Date / Time Window) Selection

Figure 1: STAR-CITY Spatio-Temporal Initialization (color print)

Bologna Italy, Miami USA and Rio Brazil, STAR-CITY can scale efficiently to any other city, which exposes real, historical together with live and heterogeneous sensors data any kind. This paper aims at presenting the in-use system description of the IBM research asset STAR-CITY. It is worth noting that the paper reflects the description of the research asset (April 2014), and not the IBM product where various components of the architecture have been re-considered for IBM products alignment. For instance IBM DB2 RDF is considered instead of Jena TDB.

This paper is organized as follows. Section 2 sketches in-use scenarios for STAR-CITY. Section 3 gives a high-level presentation of technology of the system. Section 4 highlights the lessons learned from deploying and using STAR-CITY, which is based on Semantic Web technologies. Finally Section 5 draws some conclusions and talks about future directions.

2. STAR-CITY In-use Scenarios

The STAR-CITY system is illustrated through a list of scenarios, where each highlights actions that any city manager is required to perform on a daily basis. Its scenarios have been defined with city transportation departments to support actions which are not easily supported by state-of-the-art systems in place (due to the complexity of data integration and contextual semantic reasoning). The use of semantic web technologies in all scenarios is transparent to end-users. However such technologies are strongly required to compile and deliver contextual analysis, diagnosis, exploration and prediction. All user interactions (UI) are achieved through simple UI paradigms e.g., spatial and temporal selection for initialization (Figure 1.(a) and Figure 1.(b)). All results, delivered by analysis, diagnosis, exploration and prediction, are dynamically exported as parallel, spider, pie, graph-based and time-series charts. Videos of STAR-CITY are available at http://goo.gl/TuwNyL as guidelines on how to operate the live system following the next scenarios.

2.1. *Spatio-Temporal Analysis of Traffic Conditions*

Traffic managers are interested in both historical and real-time information of traffic conditions (discretized as free, low, moderate, heavy, stopped flow) in order to extract the pulse of the city traffic at any time and space. In a context of real-time information, stream journey times (i.e., travel time estimation between fixed points in Dublin Ireland cf. Table 1) data needs to be processed in real-time while fast aggregation (average, max, min) is required for historical analysis of traffic status. In both contexts rules-based mechanisms are required to capture and infer traffic status. The STAR-CITY approach consists of discretizing numerical values of travel time individuals (described through road, link, direction, sensors) in status through SWRL[5] rules (OWL $\mathcal{EL}^{++}$ ontologies and associated rules available[6]). Figure 2 embeds the results in a parallel chart, where the status of each road segment together with its proportion (pie chart on the right hand corner) are established. While the approach is scalable for real-time status under moderate temporal intervals (up to twenty weeks), the search and aggregation over tens of months is more challenging.

2.2. *Spatio-Temporal Diagnosis of Traffic Status*

How to identify the nature and cause of traffic congestion in real-time? How to capture diagnosis results on a spatial and (historical) temporal basis? How to understand the impact of city events (road works, accident, conference, music event) on traffic conditions?

[5] http://www.w3.org/Submission/SWRL/

[6] https://ibm.biz/BdDZ5J

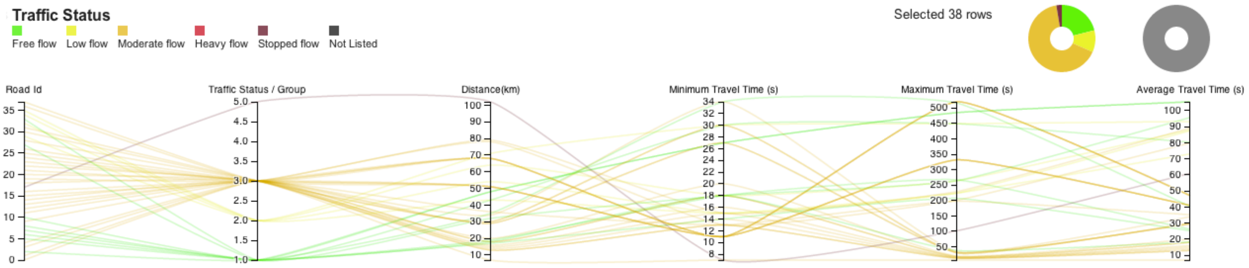


Figure 2: Spatio-Temporal Analysis of Traffic Status (color print).

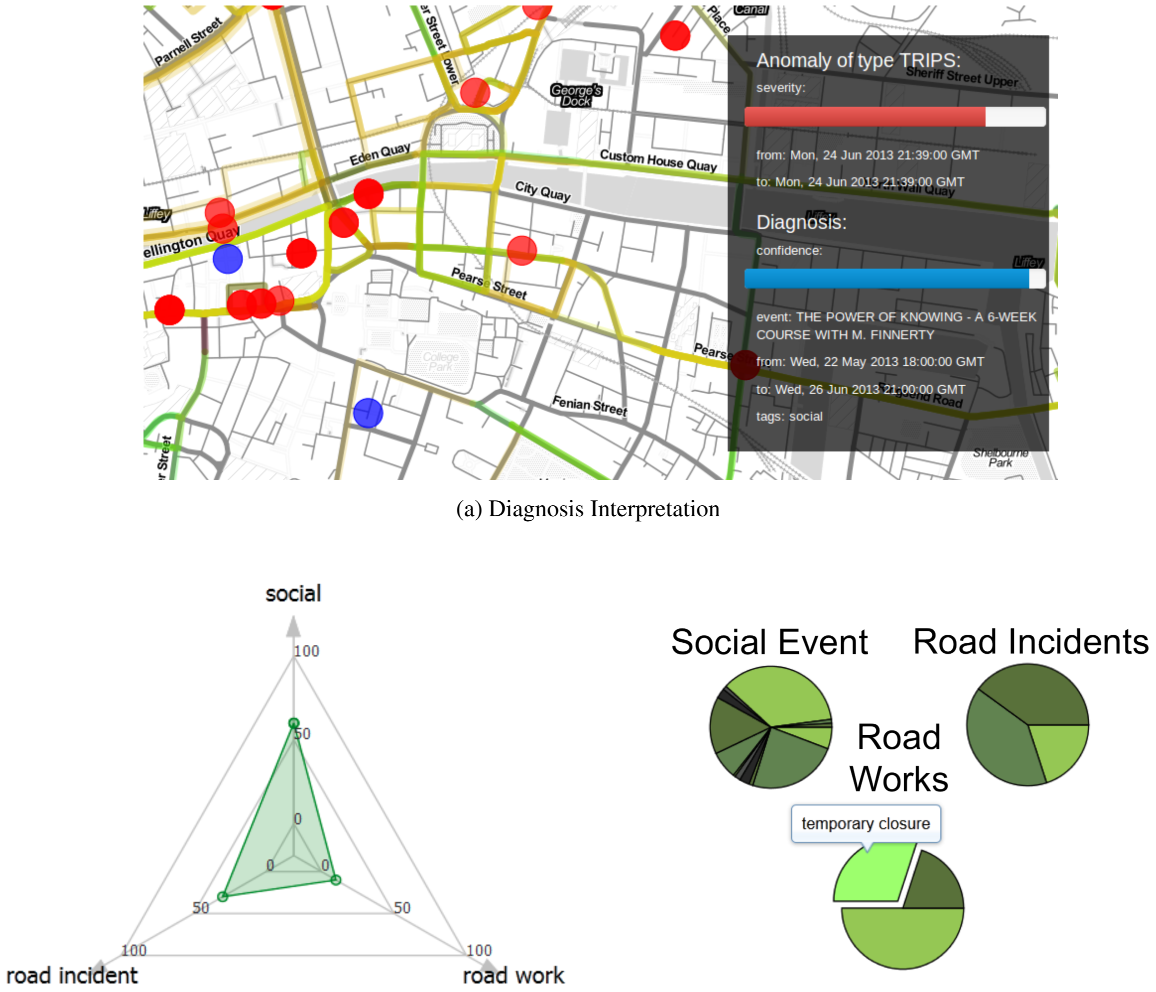
These are general questions which cannot be answered by existing state-of-the-art traffic systems, but of really importance for city managers to better understand and plan her/his cities at any time. Such question remains open because (i) relevant data sets (e.g., road works, city events), (ii) their correlation (e.g., road works and city events connected to the same city area) and (iii) historical traffic conditions (e.g., road works and congestion in Canal street on July 24th, 2013) are not fully open and jointly exploited. STAR-CITY exploits the DL-based semantics of streams to tackle these challenges. Based on an analysis of stream behavior through change and inconsistency over DL axioms, we tackled change diagnosis by determining and constructing a comprehensive view on potential causes of changes [6, 7]. Some extensions of the latter work have been achieved to support both scalable real-time and historical aggregation of diagnosis results. In addition to a spatial representation of traffic conditions and their diagnosis (Figure 3.(a)), STAR-CITY exposes a spider chart of congestion diagnosis (Figure 3.(b)), and a more in-depth analysis of all causes (Figure 3.(c)), both for any spatio-temporal constraint. More specially Figure 3.(c) captures all types (respectively subtypes) of city events which negatively impact (i.e., diagnose) the traffic conditions. All results in Figures 3.(a), (b) and (c) can be interpreted by city managers to understand how traffic condition is impacted by any type of city event. Since the diagnosis reasoning of STAR-CITY strongly relies on classification of OWL 2 EL ontologies, we adopt a distributed classification [10] of OWL 2 EL journey times individuals to obtain a scalable diagnosis. The current implementation is limited to $\mathcal{EL}^{++}$ expressivity for scalability reasons.

2.3. *Spatio-Temporal Exploration of Traffic Contexts*

The STAR-CITY system enables the city traffic managers to explore contextual information related to some city events and traffic conditions over historical semantic city data. It helps in terms of giving more insights about the city by retrieving the relevant information from the large city data to find (semantically) similar city events and their impact on previous traffic conditions. Retrieving the relevant contextual information over the heterogeneous and vast city data is a challenging task since classical search techniques are limited in terms of (i) identifying the information needs of the city managers, (ii) handling the contextual information to find similar settings happened in the past, and (iii) utilizing the heterogeneous and semantic data to retrieve accurate information. STAR-CITY addresses these issues following semantic search technologies [11, 8] and extends them significantly to handle both the context and spatio-temporal dimensions. By capturing the current context from the system (spatial, temporal, events, traffic conditions), the system formulates a contextual semantic query which better identifies the actual information need of the city managers within a certain traffic status and city setting. Then, it retrieves the relevant information (e.g., events, traffic conditions) that occurred in a similar context by using its underlying semantic search engine and displays the search results in an exploration interface. This gives the city traffic managers, for example, to get more insight about the similar events in Canal street (or of close proximity) in a similar time of the year and their profound effect on traffic conditions.

2.4. *Traffic Status Prediction*

Prediction, or the problem of estimating future observations given some historical information, is an important inference task required by city traffic managers for obtaining insight on cities. On the one hand it determines the future states of roads segments, which will support transportation departments and their managers to proactively manage the traffic before congestion is reached e.g., changing traffic light strategy. While predictive analytics spans many research fields, from Statistics, Signal Processing to Database and Artificial Intelligence [12], all existing approaches have been mainly designed for very fast processing and mining of (syntactic and numerical) raw data from sensors.


Anomaly of type TRIPS:
severity:
from: Mon, 24 Jun 2013 21:39:00 GMT
to: Mon, 24 Jun 2013 21:39:00 GMT
Diagnosis:
confidence:
event: THE POWER OF KNOWING - A 6-WEEK COURSE WITH M. FINNERTY
from: Wed, 22 May 2013 18:00:00 GMT
to: Wed, 26 Jun 2013 21:00:00 GMT
tags: social

(a) Diagnosis Interpretation

(b) Diagnosis Analysis

(c) In Depth Diagnosis

Figure 3: Spatio-Temporal Historical and Real-Time Diagnosis in STAR-CITY (color print)

However they rarely utilize exogenous sources of information for adjusting estimated prediction. Inclement weather condition, a concert event, a car accident, peak hours are examples of external factors that strongly impact traffic flow and congestion [13]. They also all fail in using and interpreting underlying semantics of data, making prediction not as accurate and consistent as it could be, specially when data streams are characterized by texts or sudden changes over time. STAR-CITY shows that the integration of numerous sensors, which expose heterogenous, exogenous and raw data streams such as weather information, road works, city events or incidents is a way forward to improve accuracy and consistency of traffic congestion prediction [9, 14]. Figure 4 illustrates how predictions are handled in STAR-CITY. The future status of road segments (in the selected boundary box) and their proportion are reported up to five hours ahead.

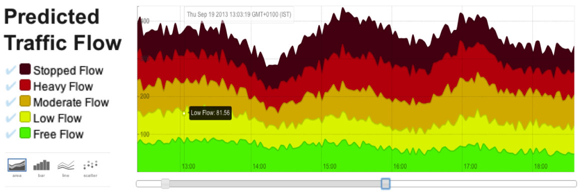


Figure 4: Prediction of Traffic Status (color print).

Similarly to diagnosis reasoning, the scalability of predictive reasoning is highly coupled with the polynomial-time characteristics of subsumption-based reasoning in DL $\mathcal{EL}^{++}$.

3. STAR-CITY Technologies

This section gives a high-level presentation of the main technologies behind STAR-CITY with a particular focus on their semantic characteristics.

3.1. Semantic Representation

Semantic technologies were used to compare and evaluate different context e.g., events (and their properties: venue, category, size, types and their subtypes), weather information (highly, moderate, low windy, rainy; good, moderate, bad weather condition). More importantly they were required for (automatically) designing, learning, applying rules at reasoning time for analysis, diagnosis and prediction components. At the end, all interfaces of STAR-CITY produces and consumes semantic representation of data. All interactions of STAR-CITY are possible because of the semantic engine, which runs behind the scene. For instance, the spatio-temporal exploration of diagnosis is only possible if the underlying data is described with semantics.

The model we consider to represent static background knowledge and semantics of data stream is provided by an ontology, encoded in OWL 2 EL[7]. The selection of the W3C standard OWL 2 EL profile has been guided by (i) the expressivity which was required to model semantics of data in our application domain (cf. Table 1), (ii) the scalability of the underlying basic reasoning mechanisms we needed in our stream context e.g., subsumption in OWL 2 EL is in PTIME [15].

3.2. Semantic Enrichment

All raw data streams in Table 1 are served as real-time OWL 2 EL ontology streams (i.e., stream of semantic-encoded data) by using IBM InfoSphere Streams [16]. Different mapping strategies are used depending on the data format. For instance XSLT for XML, Typifier [17] for tweeter feeds or custom OWL 2 EL mapping for CSV have been used. Figure 5 describes the architecture for generating OWL EL ontology streams.

All the ontology streams have the same static background knowledge to capture time (W3C Time Ontology[8]), space (W3C Geo Ontology[9]) but differ only in some domain-related vocabularies e.g., traffic flow type, weather phenomenon, event type. These ontologies have been mainly used for enriching raw data, facilitating its integration, comparison, and matching. The DBpedia vocabulary has been used for cross-referencing entities.

The main benefits of packaging our approach using stream processing are: (i) easy synchronization of streams (with different frequency updates) and their OWL2 EL transformation, (ii) flexible and scalable composition of stream operations (e.g., transformation, aggregation, filtering) by adjusting its processing units, (iii) identification of patterns and rules over different time windows, (iv) possible extension to higher throughput sensors. All points are all natively supported by stream processing engines.

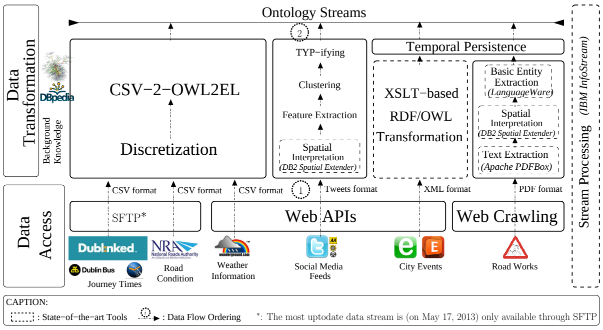


Figure 5: Semantic Stream Enrichment (color print).

The main innovation is related to the real-time transformation of stream data using a stream processing engine. STAR-CITY provides a generic mechanism for enriching real-time raw data using a pre-defined mapping descriptor.

3.3. Distributed Semantic Reasoning

The matching-based computation of context similarity, which is crucial in diagnosis and prediction components of STAR-CITY, is ensured by real-time semantic classification of ontology streams. Such classification is achieved by distributing all the standard completion OWL 2 EL rules [18] across various nodes based on their types. Each node is dedicated to at most one type of (normal form) axioms and runs its appropriate rules on axioms.

3.4. Semantic Stream Reasoning

Real-time semantic comparison and matching of stream snapshots are operated. Such computing is required by predictive reasoning and real-time diagnosis for elaborating semantic context (events, weather, incidents) similarity and correlation over time, all in real-time.

[7] http://www.w3.org/TR/owl2-profiles/
[8] http://www.w3.org/TR/owl-time/
[9] http://www.w3.org/2003/01/geo/

3.5. Diagnosis Reasoning

STAR-CITY exploits the semantics of streams exposed by sensors and its underlying data. It compiles off-line all historic diagnosis information into a deterministic finite state machine, following the structure of Dublin road network (using linkgeodata.org). The network is used (i) to properly connect roads and (ii) for exploiting congestion propagation. The state machine is augmented with respect to all semantic-augmented city events and weather conditions where a subset of them are connected to past congestion and the probability with which they have caused it. The real-time diagnosis of traffic condition is performed by analyzing the historical versions of the state machines to retrieve similar contexts i.e., congestion, potential explanation (i.e., city events), weather information. The similarity is estimated by comparing semantic descriptions of the context.

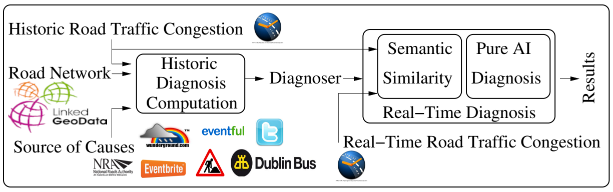


Figure 6: Diagnosis Approach Overview.

STAR-CITY extends [6] by supporting both scalable real-time and historical aggregation of traffic congestion anomalies and its diagnosis results (Figures 3.(b), (c)).

3.6. Predictive Reasoning

Once all data in Table 1 is semantically exposed (Sections 3.1, 3.2), advanced reasoning techniques are required to capture (i) changes between ontology stream snapshots, and (ii) associations of knowledge at cross-stream level, all on a time basis. The detection of changes supports the understanding of stream evolution, and then provides the basics to compute knowledge auto-correlation along a stream over time. Auto-correlation and association are core reasoning for evaluating potential patterns at one or multi-stream level(s), which are required for predicting severity of congestion. Auto-correlation evaluates semantic similarity of stream snapshots while association aims at deriving rules across streams. E.g., identifying that *"the traffic flow is never stopped on week nights in Dublin 15"* or "*a concert event is always associated with a heavy traffic flow*" are useful facts for prediction purposes.

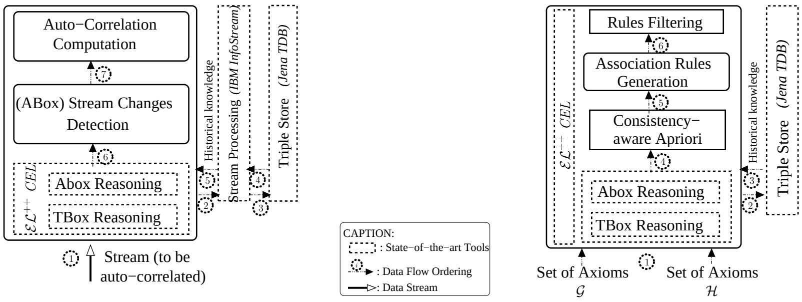
On the one hand the TBox (i.e., terminological box containing concepts and their relations) of our static background knowledge, which does not change over time, is classified once using $\mathcal{EL}^{++}$ completion rules [18]. On the other hand the ABox axioms (i.e., relations between individuals and concepts), which are generated by the ontology stream conversion (Figure 5), are internalized into TBox axioms so (i) completion rules can be applied on both axioms, (ii) TBox reasoning (e.g., subsumption, satifiablility) can be performed on internalized ABox axioms.

Cross-stream association is modeled through DL $\mathcal{EL}^{++}$ rules [19], which extends the DL $\mathcal{EL}^{++}$ expressivity while preserving its polynomial complexity. Intuitively, DL rules are encoded using SWRL rules, which is largely based on RuleML. One could, for example, formulate the timeless rule (1) "*the traffic flow of road r_1 is heavy if r_1 is adjacent to a road r_2 where an accident occurs and the humidity is optimum*". This rule connects the *journey times*, *social media* and *weather information* streams.

$$\begin{aligned} HeavyTrafficFlow(s) \leftarrow & Road(r_1) \wedge Road(r_2) \wedge \\ & isAdjacentTo(r_1, r_2) \wedge \\ & hasTravelTimeStatus(r_1, s) \wedge \\ & hasWeatherPhenomenon(r_1, w) \wedge \\ & OptimunHumidity(w) \wedge \\ & hasTrafficPhenomenon(r_2, a) \wedge \\ & RoadTrafficAccident(a) \end{aligned} \tag{1}$$

The auto-correlation of snapshots along an ontology stream is systematized in Figure 7.(a). It ensures context-aware prediction. We established it by comparing the number of changes i.e., *new*, *obsolete*, *invariant* ABox entailments between snapshots. The number of invariants entailments has a strong and positive influence on auto-correlation. On the contrary, the number of new and obsolete ABox entailments, capturing some differentiators in knowledge evolution, has a negative impact and favors negative auto-correlation.

The generation of association rules (Figure 7.(b)) between streams (and their snapshots) such as (1) is automatic and based on a DL extension of Apriori [20], aiming at supporting subsumption for determining association rules. Contrary to the initial version of Apriori, the association is achieved between any ABox elements together with their entailments (e.g., all congested roads, weather, works, incidents, city events, delayed buses). The association is possible only in the case their elements appear in at least one point of time of the streams. As the number of rules grows exponentially with the number of ABox elements and entailments in streams, we do not mine all potential rules, but filter them by adapting the definition of *support* (i.e., number of oc-


Auto−Correlation Computation
(ABox) Stream Changes Detection
Abox Reasoning
TBox Reasoning
$\mathcal{EL}^{++}$ CEL
Historical knowledge
Stream Processing (IBM InfoStream)
Triple Store (Jena TDB)
Stream (to be auto−correlated)
CAPTION:
: State−of−the−art Tools
: Data Flow Ordering
: Data Stream
Rules Filtering
Association Rules Generation
Consistency−aware Apriori
Set of Axioms $\mathcal{G}$
Set of Axioms $\mathcal{H}$

(a) Auto−Correlation (b) Association Rules Generation

Figure 7: Stream Auto-Correlation and Association Rules for Prediction.

currences that support the elements of the rule) and *confidence* (i.e., probability of finding the consequent of the rule in the streams given the antecedents of the rule) [20] for ontology stream. Rules are encoded in SWRL, and all consequents of each rule are validated though consistency checking. This ensures to obtain consistent, accurate prediction results. The number of rules is variable, mainly depending on the number of streams and complexity of their representation. For instance from $1,000$ to $5,000$ rules in a context of 2 and 7 streams respectively, cf. [14] for details.

Figure 8 presents how auto-correlation is combined with association rule generation for deriving the most relevant rules among streams.

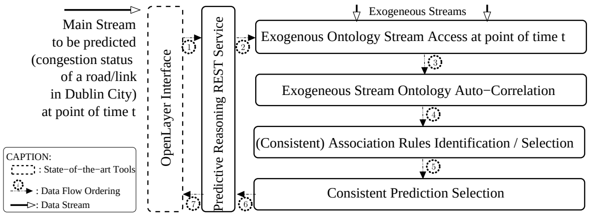

Main Stream to be predicted (congestion status of a road/link in Dublin City) at point of time t
CAPTION:
: State−of−the−art Tools
: Data Flow Ordering
: Data Stream
OpenLayer Interface
Predictive Reasoning REST Service
Exogeneous Streams
Exogenous Ontology Stream Access at point of time t
Exogeneous Stream Ontology Auto−Correlation
(Consistent) Association Rules Identification / Selection
Consistent Prediction Selection

Figure 8: Scalable and Consistent Prediction.

We first identify the context (e.g., *mild weather*, *road closure*) where the prediction is required, and then perform its auto-correlation with historical contexts. Then, we identify and select rules based on their support, confidence and consistency, but only if the consequent of the rule is consistent with the knowledge captured by the exogenous stream. The significance of rules is contextualized and evaluated against only auto-correlated stream snapshots. Thus, the selection of rules, following [9], is driven by auto-correlation, making the selection knowledge evolution-aware. This ensures to learn rules that could be applied in similar contexts i.e., where knowledge does not drastically change.

3.7. Web-based Application

This section describes the web-based application component of the STAR-CITY application.

3.7.1. REST Interface

All functionalities of STAR-CITY are exposed through REST services, providing component-ization, evolve-ability via loose coupling and hypertext.

3.7.2. Web User Interface

STAR-CITY strongly relies on HTML, CSS, Javascript (Dojo toolkit, D3, JQuery libraries) to produce an appealing user interface. Time-series, spider charts together with parallel charts are examples where Dojo and D3 components were combined with HTML and CSS.

3.7.3. Deployment

Our technology stack is based on well-established commercial components from IBM e.g., IBM InfoSphere Streams [16] for stream enrichment and processing, IBM WebSphere as the HTTP/Application Server. STAR-CITY is also based on state-of-the-art components such as OpenLayers[10] which is an open source JavaScript library for displaying dynamic map data, pssh for parallel distribution of reasoning, Jena TDB[11] as RDF store. We preferred the B+ Trees indexing structures which scale better in our context of many (stream) updates.

4. Lessons Learned

This section draws the main lessons learnt for having developed and experimented a system based on Semantic Web technologies which has been deployed in different contexts i.e., cities.

[10] http://openlayers.org/

[11] http://jena.apache.org/documentation/tdb/index.html

4.1. Heterogeneous Context and Sensors

Since the content of each data stream (sensor) source does not (usually) change, it is crucial to identify the most appropriate vocabulary, with the appropriate expressivity to represent its semantics. We mainly used DBpedia and various (W3C, NASA) standard ontologies for linkage, integration and interoperation with externals source of data. However, some cases (i.e., journey times data) required more specific and fine-grained descriptions with higher expressivity for matching and reasoning purposes. To this end we carefully developed our own terminologies, aligned with the schema of raw data, for reusability and reasoning. The on-the-fly integration with other data sources could be complex as the terminology needs to (i) be aligned with the ones which are already in place, and (ii) support relevant reasoning cross-stream data sources.

4.2. Semantic Expressivity

The current implementation is limited to OWL EL[12] as semantic encoding of city events for the computation of semantic similarity. The OWL 2 EL profile is designed as a subset of OWL 2 that is suitable in our context since ontology classification can be decided in polynomial time, hence scalable. The reasoning mechanisms in Figure 7 are highly coupled with the polynomial-time CEL reasoner for determining subsumption and consistency, which fits OWL 2 EL.

However considering OWL 2 Full or DL could have (i) triggered more causes for road congestions, and improved the diagnosis precision, and (ii) triggered stronger rules while reducing its number, hence improving the scalability (to some extent) and accuracy of prediction. It would be also interesting to evaluate the impact of using a subset of $\mathcal{EL}^{++}$ on the computation performance and the diagnosis and prediction results. In other words, which *expressiveness does fit the better for this application* is still an open challenge. Some experiments are required to provide the most appropriate context and trade-off complexity and expressivity.

4.3. Scalability

Jena TDB failed to correctly handle simultaneous updates (coming from various streams). Thus the ontology stream needs to be slightly desynchronized from each other to ensure that Jena TDB handles correctly its transaction model. To this end we simply delayed some of the streams to obtain a sequence of updates instead. We ensure such a desynchronization through our stream processing platform. The B+Trees indexing structure of TDB scales the best in our stream context where large amounts of updates are performed i.e., the transaction model is much better handled in this structure. However there were some scalability issues to handle historical data over more than approximately 110 days. If we do not limit in space and time, and if we do not apply some heuristics (e.g., by restricting to a few days of historic) we could end-up dealing with $3,800,000+$ events (in a - not worst case - context of 458 days of data, where data is updated every 20 seconds). If we consider bus status that is multiplied by $1,000$ i.e., the number of buses.

Some challenges such as data and knowledge summarization, stream synchronization are then important challenges that need to be tackled, as they both limit the scalability of the approach to some extent.

4.4. Noisy Sensor Data

In the real world, sensors exhibit noise i.e., they do not observe the world perfectly. The causes range from malfunctioning, mis-calibration, to network issues and attrition breakdown. Noisy data needs to be detected early to avoid a useless semantic enrichment, which could raise to more important problems at reasoning time, reaching to completely inaccurate diagnosis or prediction (due to alteration of rules support and confidence). We partially addressed this problem by integrating some *custom filter operators* at stream processing level to check validity of data e.g., data range checking, exceptions. The filter operators are defined by analyzing historical data of sensors. The minimum and maximum values emitted by each sensor have been computed. All records which strongly deviate from this interval are automatically removed. The integration of new data stream needs a careful analysis of historical data in order to identify the most appropriate filters, avoiding as much noise as possible.

4.5. Temporal Reasoning

Data streams evolve over time, and release new snapshots at various point of time, making the data stream integration complex. We considered the W3C Time ontology to represent the starting date/time and the duration of each snapshot, the temporal similarity is strictly based on time intervals. In other words the exploration is achieved on all city events and anomalies that meet this time interval. Some refinements of our approach are required to capture more generic and flexible temporal aspects such as anomalies and diagnosis during *rush hours*, *bank holidays*, *week-end*. Other more complex time feature could have been used e.g., temporal

[12] http://www.w3.org/TR/owl2-profiles/

intervals. This would support more complex reasoning to reason over time. For scalability reasons we use basic methods to evaluate loose temporal similarity and then integrate data stream at time level. However research challenges, already tackled by [21], would need to be considered for more accurate temporal joints.

5. Conclusion

We presented STAR-CITY, an innovative system which has been designed for (i) smoothly aggregating heterogeneous real-time data and (ii) delivering contextual analysis, diagnosis, exploration and prediction of traffic conditions in Dublin Ireland while being scalable to any city and contexts that involve sensor data. Bologna Italy, Miami USA and Rio Brazil are other examples of cities where the system has been experimented. STAR-CITY delivers insight to interpret historical and real-time traffic conditions, making road traffic easier to be managed and supporting efficient urban planning. Thus STAR-CITY supports city managers in understanding effects of city events on traffic conditions in order to take corrective actions. Semantic web technology stack has been deeply used for describing, integrating and reasoning over heterogeneous city sensor data. The experiments, detailed in [6, 14], have shown scalable, accurate and consistent analysis, diagnosis, exploration and prediction of road traffic conditions, main benefits of the semantic encoding and underlying reasoning. This paper has also highlighted the lessons learned from deploying and using a system based on Semantic Web technologies.

As emphasized in Section 4, handling (i) data summarization, (ii) flexible data integration, (iii) advanced temporal similarity, (iv) noisy data streams, among others, are future domains of investigation.

Acknowledgement

The research leading to these results has received funding from the European Unions Seventh Framework Programme (FP7/2007-2013) under grant agreement ID 318201 (SIMPLI-CITY).

References

[1] D. Schrank, B. Eisele, T. Lomax, 2012 urban mobility report, `http://goo.gl/Ke2xU` (2012).

[2] R. Arnott, K. Small, The economics of traffic congestion, American Scientist (1994) 446–455.

[3] T. Lajunen, D. Parker, H. Summala, Does traffic congestion increase driver aggression?, Transportation Research Part F: Traffic Psychology and Behaviour 2 (4) (1999) 225–236.

[4] M. Bando, K. Hasebe, A. Nakayama, A. Shibata, Y. Sugiyama, Dynamical model of traffic congestion and numerical simulation, Physical Review E 51 (1995) 1035–1042.

[5] T. Nadeem, S. Dashtinezhad, C. Liao, L. Iftode, Trafficview: traffic data dissemination using car-to-car communication, ACM SIGMOBILE Mobile Computing and Communications Review 8 (3) (2004) 6–19.

[6] F. Lécué, A. Schumann, M. L. Sbodio, Applying semantic web technologies for diagnosing road traffic congestions, in: International Semantic Web Conference (2), 2012, pp. 114–130.

[7] F. Lécué, Diagnosing changes in an ontology stream, in: AAAI, 2012.

[8] V. Bicer, T. Tran, A. Abecker, R. Nedkov, Koios: Utilizing semantic search for easy-access and visualization of structured environmental data, in: ISWC (2), 2011, pp. 1–16.

[9] F. Lécué, J. Z. Pan, Predicting knowledge in an ontology stream, in: IJCAI, 2013.

[10] R. Mutharaju, Very large scale owl reasoning through distributed computation, in: International Semantic Web Conference (2), 2012, pp. 407–414.

[11] V. Bicer, T. Tran, R. Nedkov, Ranking support for keyword search on structured data using relevance models, in: CIKM, 2011, pp. 1669–1678.

[12] H. Wang, W. Fan, P. S. Yu, J. Han, Mining concept-drifting data streams using ensemble classifiers, in: KDD, 2003, pp. 226–235.

[13] S. Cairns, C. Hass-Klau, P. Goodwin, Traffic impact of highway capacity reductions: Assessment of the evidence, Landor Publishing, 1998.

[14] F. Lecue, R. Tucker, V. Bicer, P. Tommasi, S. Tallevi-Diotallevi, M. Sbodio, Predicting severity of road traffic congestion using semantic web technologies, in: Extended Semantic Web Conference, 2014, pp. 611–627.

[15] C. Haase, C. Lutz, Complexity of subsumption in the [escr][lscr] family of description logics: Acyclic and cyclic tboxes, in: ECAI, 2008, pp. 25–29.

[16] A. Biem, E. Bouillet, H. Feng, A. Ranganathan, A. Riabov, O. Verscheure, H. N. Koutsopoulos, C. Moran, Ibm infosphere streams for scalable, real-time, intelligent transportation services, in: SIGMOD, 2010, pp. 1093–1104.

[17] Y. Ma, T. Tran, V. Bicer, Typifier: Inferring the type semantics of structured data, in: International Conference on Data Engineering (ICDE), 2013, pp. 206–217.

[18] F. Baader, S. Brandt, C. Lutz, Pushing the el envelope, in: IJCAI, 2005, pp. 364–369.

[19] M. Krötzsch, S. Rudolph, P. Hitzler, Description logic rules, in: ECAI, 2008, pp. 80–84.

[20] R. Agrawal, R. Srikant, Fast algorithms for mining association rules in large databases, in: VLDB, 1994, pp. 487–499.

[21] C. Lutz, Interval-based temporal reasoning with general tboxes, in: IJCAI, 2001, pp. 89–96.