Applying Semantic Web Technologies for Diagnosing Road Traffic Congestions

Freddy Lecue, Anika Schumann, Marco Luca Sbodio

IBM Research, Smarter Cities Technology Centre
Damastown Industrial Estate, Dublin, Ireland
{(firstname.lastname)@ie.ibm.com}

Abstract. Diagnosis, or the method to connect causes to its effects, is an important reasoning task for obtaining insight on cities and reaching the concept of sustainable and smarter cities that is envisioned nowadays. This paper, focusing on transportation and its road traffic, presents how road traffic congestions can be detected and diagnosed in quasi real-time. We adapt pure Artificial Intelligence diagnosis techniques to fully exploit knowledge which is captured through relevant semantics-augmented stream and static data from various domains. Our prototype of semantic-aware diagnosis of road traffic congestions, experimented in Dublin Ireland, works efficiently with large, heterogeneous information sources and delivers value-added services to citizens and city managers in quasi real-time.

1 Introduction

Consider the case of city planning in anticipation of a large event that requires city-wide mobilization of urban resources - a key Republic of Ireland World Cup qualifier match in Croke Park, for example. By integrating and correlating partial observations from multiple data sources, we could infer that the unseasonable inclement weather, coupled with 83,000 people descending on one area in Dublin to watch a mid-week match on a normal working day, coupled with a lack of public parking, led to traffic chaos that was widely reported in the media, driving strong negative sentiment towards the handling of such events. Whilst such an analysis is a useful forensic tool for understanding *what went wrong* and *what were the causes* **after the event**, it is important to compute causes of such unexpected situations **in quasi real-time**. This ensures that city managers have a solid understanding of the issues that lead to an unexpected situation, and can then take appropriate corrective actions.

Even if traffic congestion can be easily detected, visualized and analyzed [1] through stream data and optimization mechanisms using existing data mining [2] and machine learning approaches [3], (i) *explaining* their causes, (ii) *predicting* their impact and (iii) *recommending* alternative solutions are more complex and challenging problems, mainly due to the lack of information, its correlation and interpretation. This work focuses on the former problem, also known as *diagnosis* i.e., identification of the nature and cause of road traffic congestion.

What could be the cause of a motorway traffic congestion? Is it broken traffic light, an accident, a large concert, some road works, a brief stall, a temporarily overcrowded highway entrance or exit? The latter are potential causes of road congestions which could happen in a city traffic. Unfortunately, it is not always straightforward to obtain clear and descriptive explanations on their reasons, especially in quasi real-time situations. Understanding potential causes is important for informing interested parties, for instance, car drivers and public authorities, in quasi real time. This is important not only for providing explanations to drivers who are sitting in bumper-to-bumper traffic, but also for ensuring that public authorities will take optimal decisions and appropriate actions (e.g., rerouting or changing traffic light strategy in case of an accident or a broken traffic light) in time, especially in case of emergency.

How do large events such as a concert could impact traffic conditions? Shall we expect delays? Is re-routing appropriate? Such questions remain open because (i) relevant data sets (e.g., road works, city events), (ii) their interlinking (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, 2010) are not fully and jointly exploited. Pure AI diagnosis approaches focus on point (iii) for inferring the cause-effect relationships while semantic web technologies tackle (i) and (ii) for integrating heterogeneous and large data. This work extends the scope of pure AI diagnosis approaches to compute accurate diagnoses for situations where cause-effect relationships have not been established before. The list of potential heterogeneous sources of effects (road traffic congestion) and their causes (e.g., road weather conditions, events) that we consider in our scenario are listed in Table 1. A large part of data is provided by DCC (Dublin City Council) through *dublinked.ie*[1]agreement, and hosted at IBM.

We applied semantic web technologies for integrating heterogeneous data and then enabling advanced analytics. We exploit static and stream data (stream for short) from the road traffic data by encoding their semantics using existing LOD vocabularies (*Semantic Data* column of Table 1) and ontologies we developed for missing concepts. Road congestions are captured by correlating Dublin City Bus streams with latter light-weight ontologies. Diagnosis results are retrieved and interpreted by exploiting semantic representation of historical data and infrastructure data such as road network and bus lines. The quasi real-time cause-effect analysis is then reported back to the users.

The remainder of this paper is organized as follows. In Section 2 we present how road congestions are captured. Section 3 presents our semantics-augmented approach for diagnosing road congestions. Section 4 presents details about the prototype implementation and reports some experiment results regarding its applicability and scalability. Section 5 briefly comments on related work. Section 6 draws some conclusions and talks about possible future directions.

2 Detecting Road Traffic Congestion using Semantics of Stream

The model we consider to represent static background knowledge and semantics of stream data (a.k.a. evolving knowledge over time) is provided by an ontology. Dynamic knowledge is then captured by reasoning on these ontology-augmented data descriptions. We focus on W3C standard OWL 2 to represent such ontologies since its logic

[1] http://dublinked.ie/

(DL) offer good reasoning support for most of its expressive profile. This section[2] reviews (i) OWL 2 EL and its DL $\mathcal{EL}^{++}$ as a formal knowledge representation language to define (ii) ontology stream and infer (iii) road congestions. Fig.1 positions the reviewed elements in relation to our challenge: *diagnosing road congestions*.

	Data Source	Description	Format Type	Temporal Frequency (s)	Historic (mm/yyyy)	Size Estimation per day (GBytes)	Data Provider
Source of Effects	Dublin Bus	Vehicle activity (GPS location, line number, delay, stop flag)	SIRI[a] XML-based	20	11/2010	4-6	(Private) DCC
Source of Causes	Wunderground for Dublin	Real-time weather information	CSV	[5, 600] (depending on stations)	01/1996	[0.050, 1.5] (depending on stations)	(Public) Wunderground[b]
	Road Weather Condition (54 stations)		CSV	600	11/2010	0.1	(Public) NRA[c]
	Road Works and Maintenance		CSV	3600	11/2010	0.01	(Public) Dublinked[d]
	Events in Dublin	Events with small attendance	XML	Not considered	11/2011	0.001	(Public) Eventbrite[e]
		Events with large attendance			11/2011	0.05	(Public) Eventful[f]
Semantic Data	DBPedia	Structured facts extracted from wikipedia	RDF	No	No	3.5×10^6 concepts	(Public) DBPedia[g]
	Dublin City Roads (listing of type, junctions, GPS coordinate)		RDF	No	No	0.1	(Public) Linkedgeodata[h]

[a] SIRI (Service Interface for Real Time Information) is a standard for exchanging real-time information about public transport services and vehicles - http://siri.org.uk

[b] http://www.wunderground.com/weather/api - http://www.wunderground.com/history/airport/EIDW/2012/5/28/DailyHistory.html?format=1

[c] NRA - National Roads Authority http://www.nratraffic.ie/weather

[d] http://dublinked.ie - Sample: http://www.dublinked.ie/datastore/metadata064.php

[e] https://www.eventbrite.com/api

[f] http://api.eventful.com

[g] http://dbpedia.org

[h] http://linkedgeodata.org

Table 1. (Incomplete) Overview of Traffic Scenario Data sets (Dublin City Dependant).

2.1 Background: OWL 2 EL and its $\mathcal{EL}^{++}$ Description Logics

The selection of the OWL 2 EL profile has been guided by (i) the expressivity which was required to model semantics of data in our application domain (Table 1) and (ii) the complexity of its underlying reasoning e.g., subsumption in OWL 2 EL is in PTIME [4]. The DL $\mathcal{EL}^{++}$ [5] is the logic underpinning OWL 2 EL and the basis of many more expressive DL. For the sake of readability we illustrate semantic representation and reasoning using DL formalism.

A signature Σ, defined by $(\mathcal{CN}, \mathcal{RN}, \mathcal{IN})$, consists of 3 disjoint sets of (i) atomic concepts $\mathcal{CN}$, (ii) atomic roles $\mathcal{RN}$, and (iii) individuals $\mathcal{IN}$. Given a signature, the top

[2] Semantic representations are illustrated in DL to keep descriptions as concise as possible.

concept $\top$, the bottom concept $\bot$, an atomic concept A, an individual a, an atomic role r, $\mathcal{EL}^{++}$ concept expressions C and D can be composed with constructs: $\top \mid \bot \mid A \mid C \sqcap D \mid \exists r.C \mid \{a\}$. We slightly abuse the notion of atomic concepts to include $\top$, $\bot$ and nominals [6] i.e., individuals appearing in concept definitions of form $\{a\}$.

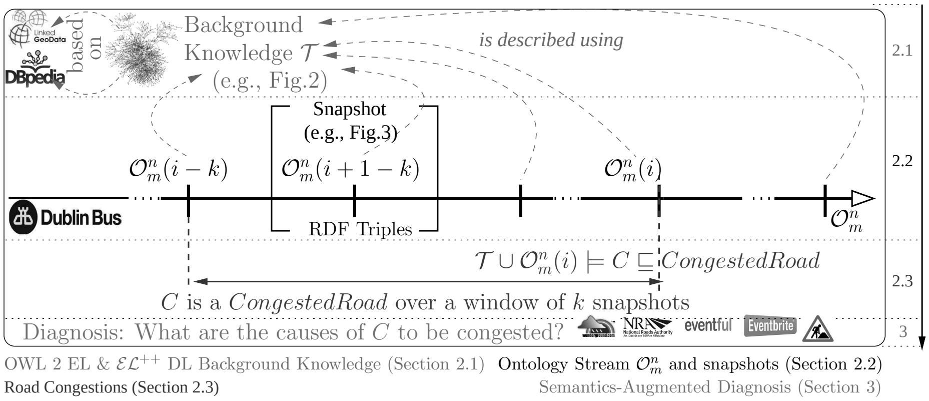
Linked GeoData
DBpedia
based on
Background Knowledge $\mathcal{T}$ (e.g., Fig.2)
is described using
Snapshot (e.g., Fig.3)
$\mathcal{O}^n_m(i-k)$
$\mathcal{O}^n_m(i+1-k)$
$\mathcal{O}^n_m(i)$
$\mathcal{O}^n_m$
Dublin Bus
RDF Triples
$\mathcal{T} \cup \mathcal{O}^n_m(i) \models C \sqsubseteq CongestedRoad$
C is a $CongestedRoad$ over a window of k snapshots
Diagnosis: What are the causes of C to be congested?
NRA National Roads Authority
eventful
Eventbrite
2.1
2.2
2.3
3
Section in the paper
OWL 2 EL & $\mathcal{EL}^{++}$ DL Background Knowledge (Section 2.1)
Ontology Stream $\mathcal{O}^n_m$ and snapshots (Section 2.2)
Road Congestions (Section 2.3)
Semantics-Augmented Diagnosis (Section 3)

Fig. 1. Road Congestions Diagnosis through Background Knowledge and Ontology Stream.

The particular DL-based ontology $\mathcal{O} \doteq < \mathcal{T}, \mathcal{A} >$, is composed of a TBox $\mathcal{T}$ and ABox $\mathcal{A}$. A TBox is a set of concept and role axioms.

$$Bus \sqsubseteq \exists id.BusID \ \sqcap \ \exists loc.GPSLocation \tag{1}$$
$$Road \sqsubseteq \exists id.RoadID \ \sqcap \ \exists in.GPSSetPoint \tag{2}$$
$$\exists id.BusID \sqcap \exists loc.(GPSLocation \sqcap \exists in.GPSSetPoint) \sqcap \exists id.RoadID \sqsubseteq Road \sqcap \exists with.Bus \tag{3}$$
$$Road \ \sqcap \ \exists with.(Bus \ \sqcap \ \exists congested.High) \sqsubseteq CongestedRoad \tag{4}$$
$$Bus \ \sqcap \ \exists delayed.High \sqsubseteq DelayedBus \tag{5}$$
$$\{r_1\} \sqsubseteq Road \ \sqcap \ \exists id.\{Canal\} \ \sqcap \ \exists roadPoint.\{(53.33, -6.27), (53.33, -6.28), (53.33, -6.29)\} \tag{6}$$

Fig. 2. Sample of an $\mathcal{EL}^{++}$ TBox $\mathcal{T}$ (GCI (6) is an internalized ABox axiom).

Example 1 ***($\mathcal{EL}^{++}$ DL Concept)***
A $CongestedRoad$ is a concept of a road with at least a congested bus (Fig.2).

$\mathcal{EL}^{++}$ supports General Concept Inclusion axioms (GCIs, e.g. $C \sqsubseteq D$ with C is subsumee and D subsumer) and role inclusion axioms (RIs, e.g., $r \sqsubseteq s$, $r_1 \circ \cdots \circ r_n \sqsubseteq s$). An ABox is a set of concept assertion axioms e.g., $a : C$, role assertion axioms e.g., $(a; b) : r$, and individual in/equality axioms e.g., $a \neq b$ or $a = b$.

We internalize ABox axioms into ($\leadsto$) TBox axioms so completion-based algorithms [5] can be applied to classify both axioms and entail subsumption. Thus TBox reasoning (subsumption, satisfiablility) can be performed on internalized ABox axioms.

$$a : C \leadsto \{a\} \sqsubseteq C \qquad\qquad (a, b) : r \leadsto \{a\} \sqsubseteq \exists r.\{b\}$$
$$a \doteq b \leadsto \{a\} \equiv \{b\} \qquad\qquad a \neq b \leadsto \{a\} \sqcap \{b\} \sqsubseteq \bot$$

Besides considering internalized ABox, we assume that $\mathcal{EL}^{++}$ TBox is normalized, and all subsumption closures are pre-computed [5]. We use the term background knowledge [7] to refer to such TBoxes.

2.2 Ontology Stream

An ontology stream [8] is considered as a sequence of ontologies (Definition 1) where knowledge is captured through its dynamic and evolutive versions.

Definition 1 ***(Ontology Stream)***
An ontology stream $\mathcal{O}_m^n$ from point of time m to point of time n is a sequence of ontologies $(\mathcal{O}_m^n(m), \mathcal{O}_m^n(m+1), \cdots, \mathcal{O}_m^n(n))$ where $m, n \in \mathbb{N}$ and $m < n$.

$\mathcal{O}_m^n(i)$ is a snapshot of an ontology stream (stream for short) $\mathcal{O}_m^n$ at point of time i, referring to a set of axioms in $\mathcal{L}$. A transition from $\mathcal{O}_m^n(i)$ to $\mathcal{O}_m^n(i+1)$ is an update.

$$\boxed{\begin{array}{lr} \mathcal{O}_0^9(6) : \{bus31\} \sqsubseteq \exists id.\{dub31\} \sqcap \exists loc.\{(53.33, -6.27)\} & (7) \\ \qquad\quad : \{bus31\} \sqsubseteq \exists congested.High & (8) \\ \mathcal{O}_0^9(7) : \{bus31\} \sqsubseteq \exists id.\{dub31\} \sqcap \exists loc.\{(53.33, -6.28)\} & (9) \\ \qquad\quad : \{bus31\} \sqsubseteq \exists delayed.High \sqcap \exists congested.High & (10) \end{array}}$$

Fig. 3. Stream Snapshots: $\mathcal{O}_0^9(6)$ and $\mathcal{O}_0^9(7)$.

Example 2 ***(Ontology Stream)***
Fig.3 illustrates a partial ontology stream $\mathcal{O}_0^9$ along $\mathcal{O}_0^9(6)$ and $\mathcal{O}_0^9(7)$. Knowledge of snapshots is captured by GCIs e.g., $\{bus31\}$ is both delayed and congested in $\mathcal{O}_0^9(7)$.

2.3 Road Congestions

Road congestions (4) are derived by first capturing dynamic knowledge from the stream ontology, where the latter is then interpreted using background knowledge. Following (4), updating the definition of road congestions is straightforward.

Example 3 ***(Road Congestions)***
Road $\{r_1\}$ is a $CongestedRoad$ in $\mathcal{O}_0^9(7)$ with respect to GCIs (1-4), (6), (9-10).

In the following we will focus on diagnosing k-invariant road congestions i.e., congestions which remain persistent over a sequence of k snapshots. The diagnosis result is then extracted from this k-window i.e., all causes should occur in this window. Therefore, snapshots to be explored for diagnosis are pre-determined. In case of overlapping snapshots, the latter are considered once to avoid duplicate information, and all k snapshots are considered without any distinction.

Example 4 ***(k-Invariant Road Congestions)***
$\{bus31\} \sqsubseteq CongestedRoad$ is a 2-invariant road congestion from $\mathcal{O}_0^9(6)$ to $\mathcal{O}_0^9(7)$.

3 Semantics-Augmented Diagnosis

Diagnosis [9] is the task of explaining anomalies (e.g., congested roads) given a flow of observations. Interpreted in the context of the *Semantic* and *Stream Web*, anomalies are

k-invariant road congestions and observations are captured from background knowledge $\mathcal{T}$ (e.g., any bus is conducted on roads) and dynamic knowledge of $\mathcal{O}_m^n$ (e.g., a bus is in a heavy traffic and a sport event is active in some snapshots). Our approach (Fig.4) elaborates an off-line diagnoser (Section 3.1) which aims at capturing *historical observations* over a *window timeframe* of k and their *explanations*. In other words diagnosis of historical anomalies is captured by the off-line diagnoser e.g., Canal street was congested in 2012, May 1^{st} at 6:00pm because of a concert event in Aviva stadium and road works in Bath avenue. Then quasi real-time diagnosis (Section 3.2) consists in retrieving "*similar*" causes (e.g., roads with heavy traffic of same duration) with "*similar*" conditions (e.g., close sport event) which have appeared in the past, and then reporting back their explanation through an interpretation in quasi real-time conditions.

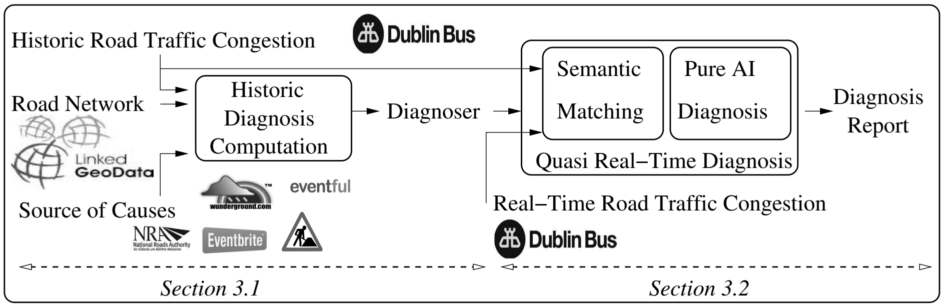

Fig. 4. Overview of the Semantics-Augmented Diagnosis Approach.

3.1 Historic Diagnosis Computation

The traffic congestion diagnoser compiles off-line all historic diagnosis information into a deterministic finite state machine. The latter state machine is retrieved with respect to all RDF-augmented events, road works, road and weather conditions where a subset of them are connected to historic traffic congestions and the probability with which they have indeed caused it. Our traffic congestion diagnoser is strongly inspired from the Dublin City road network (using linkgeodata.org and complementary information[3]) to properly connect roads and to consider congestion propagation.

Illustrated in Fig.5, a diagnoser is defined by its states which are road intersections or car park locations that are associated to nearby events/road works. The latter association is done using their GPS geolocation and following the haversine formula [10] to evaluate distances. The transitions of the diagnoser correspond to roads and each road is labeled by its historic diagnosis information where the latter is available for each snapshot of the day. Every two-way road, as bidirectional road in a city, corresponds to two roads in the diagnoser.

Example 5 Traffic Jam Diagnoser (where causes are defined as events)
Let 3 *car park locations:* $\{l_1\}$, $\{l_2\}$, $\{l_3\}$ *used by people driving to events* $\{e_1\}$, $\{e_2\}$, $\{e_3\}$. *The transition labels of diagnoser in Fig.5 show the historic diagnosis of a snapshot* s. *The label* $(\{e_3\}, 0.6)$ *of road* $\{r_7\}$ *indicates that its cause to be congested (C*

[3] http://www.dublinked.ie/datastore/metadata125.php

stands for DL concept CongestedRoad) is event $\{e_3\}$ with a probability of 0.6. The probability is computed by retrieving from all historic records the probabilities:

- *that $\{r_7\}$ was congested at snapshot s when $\{e_3\}$ took place , i.e.:*

$$p((\{r_7\} \sqsubseteq C)^s|\{e_3\}) := \frac{\textit{number of days with } \{e_3\} \textit{ and } \{r_7\} \textit{ being congested at } s}{\textit{number of days where } \{e_3\} \textit{ took place}}$$

- *that $\{r_7\}$ was congested at snapshot s when $\{e_3\}$ did not take place , i.e.:*

$$p((\{r_7\} \sqsubseteq C)^s|E\backslash\{e_3\}) := \frac{\textit{number of days without } \{e_3\} \textit{ and } \{r_7\} \textit{ being congested at } s}{\textit{number of days where } \{e_3\} \textit{ did not take place}}$$

$\{r_7\}$ was congested on 50% of the days where $\{e_3\}$ took place and on 20% of the days where $\{e_3\}$ did not take place. Thus, 20% of the congestions on $\{r_7\}$ at snapshot s cannot be connected to city events while 30% of the congestions are caused by $\{e_3\}$. Thus, once we detect that $\{r_7\}$ is indeed congested we obtain that with a probability of 0.6 it was congested because of the upcoming event $\{e_3\}$. The events e_1, e_2 have no impact on the traffic situation of r_7 because r_7 is a "cul de sac". Formally:

$$p^s_{r_7} := \frac{p((\{r_7\} \sqsubseteq C)^s|\{e_3\}) - p((\{r_7\} \sqsubseteq C)^s|\langle E \setminus \{e_3\}\rangle)}{p((\{r_7\} \sqsubseteq C)^s|\{e_3\})} \tag{11}$$

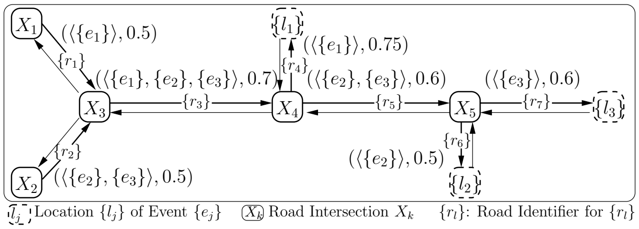
X_1
$(\langle\{e_1\}\rangle, 0.5)$
$\{r_1\}$
$(\langle\{e_1\}, \{e_2\}, \{e_3\}\rangle, 0.7)$
$\{l_1\}$
$(\langle\{e_1\}\rangle, 0.75)$
$\{r_4\}$
$(\langle\{e_2\}, \{e_3\}\rangle, 0.6)$
$(\langle\{e_3\}\rangle, 0.6)$
X_3
$\{r_3\}$
X_4
$\{r_5\}$
X_5
$\{r_7\}$
$\{l_3\}$
$\{r_2\}$
$\{r_6\}$
$(\langle\{e_2\}\rangle, 0.5)$
X_2
$(\langle\{e_2\}, \{e_3\}\rangle, 0.5)$
$\{l_2\}$
l_j Location $\{l_j\}$ of Event $\{e_j\}$
X_k Road Intersection X_k
$\{r_l\}$: Road Identifier for $\{r_l\}$

Fig. 5. A Traffic Jam Diagnoser (unlabeled roads have no causes of congestion).

We generalized the approach in a way that the traffic congestion diagnoser could handle road works, road weather and conditions as part of causes of a congested road. From a diagnosis point of view these causes are handled exactly like events $\{e_3\}$ in Example 5. From a semantic point of view this required the semantic description of road works, road weather and conditions and a corresponding matching function for determining their dis/similarity (Section 3.2) at various point of time.

3.2 Quasi Real-Time Diagnosis

As motivated in Section 1, pure AI diagnosis approaches [9, 11] are not able to retrieve any diagnosis result of quasi real-time conditions (e.g., events or road works for which we do not have any historical records) if the latter do not exactly match at least one of the existing historical conditions. So how to compute diagnosis information with respect to new conditions? We tackle this problem by means of existing semantic techniques and define a matching function $Sim_{\mathcal{T}}$ for matching new DL concept-based C_n conditions and historic conditions C_h. Conditions, defined along city events, road works, weather and road conditions, are all represented using existing vocabularies such as DBpedia, SKOS[4], Talis[5], basic geo vocabulary[6] and internal IBM ontologies for handling basic generalization/specialization of new and historic conditions. For instance road works are represented through road asset type, work description and spread while events are represented through capacity, ownership and categories. The $Sim_{\mathcal{T}}$ function is based on the matchmaking functions introduced by [12] and [13]:

- **Exact** If C_n and C_h are equivalent concepts: $\mathcal{T} \models C_n \equiv C_h$.
- **PlugIn** If C_n is sub-concept of C_h: $\mathcal{T} \models C_n \sqsubseteq C_h$.
- **Subsume** If C_n is super-concept of C_h: $\mathcal{T} \models C_h \sqsubseteq C_n$.
- **Intersection** If the intersection of C_n and C_h is satisfiable: $\mathcal{T} \not\models C_n \sqcap C_h \sqsubseteq \bot$.

All conditions C_n are matched against every C_h using the latter function so the *"similarity"* of quasi real-time and historic conditions can be evaluated. Every pair of quasi real-time and historic conditions (C_n, C_h) is then ordered based on partial ordering of matching types $Sim_{\mathcal{T}}$. The most appropriate (or semantically similar) historic conditions is then used to simulate quasi real-time conditions using the diagnoser.

In more details the quasi real-time diagnosis process is based on a traffic congestion diagnoser where quasi real-time conditions (i.e., today's event, road works and weather conditions) are approximated by historic condition using $Sim_{\mathcal{T}}$. This diagnoser is computed at the beginning of the day such that its computation time does not impact quasi real-time diagnosis. During the day, once we then detect a traffic congestion on a road at snapshot s, we use the diagnoser to look up the historic diagnosis information that explains the traffic congestion i.e. to retrieve the transition label that corresponds to the congested road at s. This information might contain conditions that do not happen today but that were only considered because of their $Sim_{\mathcal{T}}$-semantic similarity to historic conditions. In such a case we use semantic techniques for computing a diagnosis report that interprets the traffic congestion in the context of quasi real-time events.

Definition 2 ***(Diagnosis Report)***
Let $\mathcal{L}$ be a DL, $\mathcal{T}$ be a set of axioms in $\mathcal{L}$. Let C_h, C_n, C and $\{r\}$ be four concepts in $\mathcal{L}$ such that C_h and C_n are respectively historical and new conditions. Let (C_h, p_h) be historical conditions of $\{r\}$ to be congested ($\{r\} \sqsubseteq C$) with a probability p_h in $[0, 1]$. A diagnosis report $\langle \mathcal{L}, \mathcal{T}, (C_h, p_h), C_n, \{r\}, C\rangle$ of GCI $\{r\} \sqsubseteq C$ consists in finding a pair $\langle R, p \circ p_h\rangle$ where R is a concept in $\mathcal{L}$ explaining the difference between C_h and C_n, and $\circ$ is an ordering function, which positions the probability p of C_n wrt. p_h.

The diagnosis report $\langle \mathcal{L}, \mathcal{T}, (C_h, p_h), C_n, \{r\}, C\rangle$ is constructed by exploiting concept *abduction* [14] $C_n \backslash C_h$, defined by $\{B \in \mathcal{L} \mid \mathcal{T} \models C_h \sqcap B \sqsubseteq C_n\}$ between new conditions C_n and historic conditions C_h. This description $C_n \backslash C_h$ represents what is underspecified in C_h in order to completely satisfy C_n in $\mathcal{T}$. As the solution of an abduction problem is not unique, we consider the most general description B. Besides

[4] `http://www.w3.org/2004/02/skos/core`

[5] `http://schemas.talis.com/2005/address/schema`

[6] `http://www.w3.org/2003/01/geo/wgs84_pos`

retrieving similar conditions for diagnosing anomalies, concept abduction is used to report back the impact of considering non exact matching, so the estimated probability can be justified. The ordering function between p and p_h is defined based on the subsumption relation between C_n and C_h. Computing a diagnosis report is a PTIME problem due to the PTIME complexity of abduction [14] and subsumption [5] in $\mathcal{EL}^{++}$.

Example 6 ***(Diagnosis Report)***
Let $(\{e_1\}, 0.5)$ *in Fig.5 be the diagnosis result (through historical conditions) for a past traffic congestion on road* $\{r_1\}$ *and* $\{e'_1\}$ *a new event such that* $Sim_{\mathcal{T}}(\{e'_1\}, \{e_1\})$ *is PlugIn. In case of a congested road* $\{r_1\}$*,* $\{e_1\}$ *is provided as diagnosis. In addition* (12) *captures the impact of considering a PlugIn matching between* $\{e'_1\}$ *and* $\{e_1\}$*.*

$$\langle\{Event \sqcap \exists attendee.LargeType, Event \sqcap \exists attendee.Youth\},\ p \geq 0.5\rangle \qquad (12)$$

The diagnosis result reflects the real-time condition expect that `LargeType` *audience of the event has been over-generalized during diagnosis (if* $LargeType$ *is defined to be subsumed by* $SmallType$*), and that its characteristic of* $Youth$ *audience has not been considered by* $\{e_1\}$*. Since more people attend event in* $\{e'_1\}$ *than it was the case for* $\{e_1\}$ *we infer* $p \geq 0.5$*.*

4 Validation

This section reports (i) our context of experimentation, (ii) details of the prototype and (iii) a computational-based evaluation of our approach for testing its performance in real world scenarios. The main objective was to diagnose congested roads (4) using various semantics-augmented real live streams and static data in Table 1.

timestamp	line reference	direction	journey patterns reference	in congestion	latitude	longitude	delay	block reference	vehicle	point number	at stop
1322352000000	41	0	041B0002	0	-6.3026	53.4877	-182	41018	33427	3874	1
1322352000000	27	0	271003	0	-6.28932	53.3357	0	27101	33286	2190	0
1322352000000		0	null	0	-6.25923	53.3448	0	4001	33380	7226	0
1322352000000	41	0	410001	0	-6.2375	53.466	0	41013	33631	7348	0
...	...	...	...	...	...	...	...	...	...	...	...

Table 2. SIRI Data Fragment (Headers are not part of the SIRI data but here for clarity).

4.1 Context: Dublin City

(i) The **Dublin Bus Stream** is encoded according to the SIRI standard (footnote *a* in Table 1), and the real-time stream is persisted into CSV file. Each file represents one day of SIRI data i.e., information of 1000 buses is updated every 20 seconds (Table 2).

Each SIRI record (line in a CSV file) contains information about the current position (latitude and longitude) of a bus. The bus line is uniquely identified by two fields *line reference* and *journey pattern reference*. The boolean fields *direction*, *in congestion*, and *at stop* indicate respectively the bus direction along the bus line, if the bus is in congestion, and if the bus is at a stop point (*point number*). Information about line references, journey pattern references, and stop points is given separately through other CSV file; such information is static (or at least it changes very rarely). We have developed a simple $\mathcal{EL}^{++}$ ontology to represent SIRI data (DL samples in Fig.2 and 3). Fig.6 shows the main classes in the ontology. These classes model the core of the static SIRI information: line references, journey pattern references, and point numbers (bus stops). The class *InterStopDistance* is used to provide information about the distance between two point numbers along a journey pattern: following the relationships *fromPointNumber* and *toPointNumber*, it is possible to reconstruct the entire path of a bus line along with the distances between pairs of consecutive point numbers.

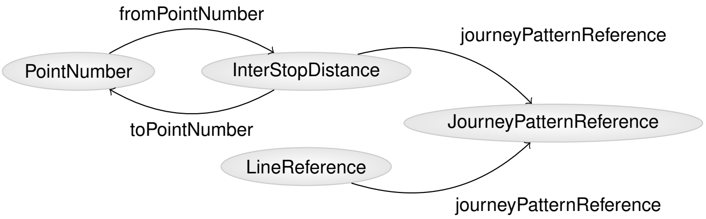

Fig. 6. Core Part of the SIRI Ontology modeling a *VehicleAtomicUpdate* Concept.

The actual SIRI records (lines from the CSV files) are modeled as instances of the class *VehicleAtomicUpdate*, which has a property for each field of a SIRI record (Table 2). Note that some of the fields of a SIRI record may lack some field values: then the instance of *VehicleAtomicUpdate* will lack the corresponding properties. Based on an history of 217 SIRI data files (approximately 26 GB), referring to 217 days in 2011 (approximately 122 MB a day), 44.7% of the SIRI records generates 8 triples and 47.2% generate 11 triples (1 triple to define the type of the RDF resource, and 10 triples to specify the SIRI properties); the other records generate either 9 or 10 triples. The varying number of triples per record is due to some missing fields, mostly the line reference and the journey pattern reference. The instances of VehicleAtomicUpdate with missing properties are nevertheless useful to estimate the number of buses in a bounding box in a certain time window (latitude, longitude, timestamp are always available).

(**ii**) **City Events** were captured through *Eventful (http://eventful.com)* and *EventBrite (http://www.eventbrite.com)* where an average of 187 events a day (i.e., same days as those captured for SIRI data) have been described using some LOD vocabularies e.g., DBpedia, Talis. In addition we enriched the events description with $\mathcal{EL}^{++}$ GCIs to capture fined descriptions of their categories. The latter has been considered for computing not only fined matching between historical and new events but also for computing the diagnosis report (Example 6). Each event has been described on average through 26 RDF triples. In this respect our event description model is not as complete as Event-F [15], but have some extensions for capturing city events (e.g., organizer, venue). The model is also more specific than LODE[7] and the Event ontology[8].

[7] `http://linkedevents.org/ontology/`

[8] `http://purl.org/NET/c4dm/event.owl#`

$(\textbf{iii})$ Similarly an average of 51 **Road Works and Maintenance**[9] records a day have also been enriched through 16 RDF triples each. An $\mathcal{EL}^{++}$ enrichment of this raw data ensures that historical and new records can be matched for diagnosis purposes. Again, the way they match is reported for diagnosis inspection.

$(\textbf{iv})$ We also injected $14,316$ $\mathcal{EL}^{++}$ GCIs (through 6 RDF triples each) to describe 4772 **Roads and their Interconnections**[10].

$(\textbf{v})$ The **Core Static Ontology**, which is used for representing SIRI, events, road works, road weather and Dublin weather data, is composed of 67 concepts with 24 role descriptions (25 concepts subsume the 42 remaining ones with a maximal depth of 4).

$(\textbf{vi})$ Finally a **History of** 217 **days of the Traffic Congestion Information** was computed based on stream bus data (encoded by more than 1×10^9 RDF triples) recorded for 217 days. Information about past events, road works, wheather information and road conditions was stored as 1.1×10^6 RDF triples. The traffic congestion diagnoser, consists of $10,856$ transitions and $4,128$ states, $4,076$ of which correspond to road intersections and 52 car parks of event locations. Every diagnoser transition had $4,320$ labels corresponding to all snapshots of a day. Each label contained 0 to 8 causes that with a probability of 0 to 0.74 have caused a traffic congestion at the particular snapshot and road.

4.2 Architecture, Implementation and Prototype

The prototype extends [1] (aiming at displaying traffic conditions in real-time) by providing explanation of road congestions. Congested roads are selectable and information about causes of such situations are displayed and refreshed in quasi real-time. Its implementation consists of (i) a RDFizer which encodes syntactic data in RDF[11], together with elements of Fig.4 i.e., (ii) road congestions detection, (iii) historic diagnosis computation and (iv) quasi real-time diagnosis.

• **On-Demand Stream Data RDFization** of the Dublin bus stream data is exposed as REST APIs (Fig.7). Its RDFization is important not only for capturing road congestions but also to identify potential source of causes. The REST API takes as input two timestamps t_i and t_f, and it generates the RDF representation of the SIRI data in the interval $[t_i, t_f]$ i.e., instances of the ontology class *VehicleAtomicUpdate* in Fig.6. Due to large amount of data, we use an *Indexer* that periodically indexes SIRI data to quickly identify the file that contains t_i. Then we perform a binary search in that file to find the line having the closest timestamp to t_i; we start reading from that line and we continue (potentially across multiple files) until we reach t_f. Then we perform a binary search in that file to find the line having the closest timestamp to t_i. While reading the SIRI data for the requested time window, we generate RDF triples describing each record, and store the triples in an RDF store; the current prototype uses Jena TDB [12]

[9] CSV sample in `http://www.dublinked.ie/datastore/metadata064.php`

[10] CSV sample in `http://www.dublinked.ie/datastore/metadata125.php`

[11] We focus on the on-demand transformation of stream SIRI data. Standard approaches - http://www.w3.org/wiki/ConverterToRdf - can been used for RDFizing static CSV, XML data.

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

as RDF store, but we are currently integrating IBM DB2 RDF store [13]. To avoid problems with possible TDB datasets corruptions, we currently create a new dataset for each SIRI-to-RDF transformation requested: the dataset will contain the static SIRI data in the default graph, and the RDF representation of the requested time window in a named graph. The REST API returns to users a unique identifier for its request, and then generates the RDF in background. The user can check when the requested transformation is completed by providing the unique identifier. We also provide a REST API for querying a dataset, supporting SPARQL SELECT, ASK and CONSTRUCT queries.

• **Road Congestion Detection** is achieved using a DL extension of InfoSphere Streams[14] [1] for real-time detection of road congestions.

• **Historic Diagnosis Computation** is done by elaborating the traffic congestion diagnoser i.e., Dublin city roads (footnote *h* in Table 1) annotated with the diagnosis results explaining historical congestions.

• **Quasi Real-Time Diagnosis:** The diagnoser [9] has been enhanced by reporting the approximation of historical and quasi real-time observations. We have implemented the semantic reasoning part using CEL DL reasoner[15] to check satisfiability, subsumption, and MAMAStng[16] to construct abduction between diagnosis.

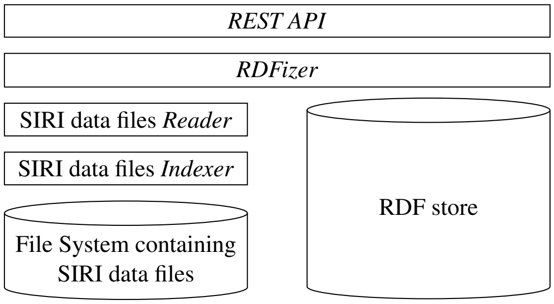

Fig. 7. Stream Data RDFization Architecture.

4.3 Experimentation

We report the computation time of (i) the overall diagnosis approach and then more specifically (ii) the semantic matching part of the diagnosis approach. Experiments were run on a server of 4 Intel(R) Xeon(R) X5650, 2.67GHz cores and 6GB RAM.

• **Overall Approach Experimentation - Context:** Fig.8 illustrates the impact of windows of exploration and size of the ontology streams on the computation time of elements in Section 4.2. The window of exploration of the dynamic knowledge is experimented from a range of 1 min (i.e., 3 snapshots) to 20 min (i.e., 60 snapshots). Besides axioms about bus information, C_1 captures all stream information i.e., road works, events, road weather and weather condition (Section 4.1) while C_2 only captures axioms about bus information and road works i.e., 83% of axioms.

[13] http://www-01.ibm.com/software/data/db2-warehouse-10/
[14] http://www-01.ibm.com/software/data/infosphere/streams/
[15] http://lat.inf.tu-dresden.de/systems/cel
[16] http://dee-227.poliba.it:8080/MA-MAS-tng/DIG

• **Overall Approach Experimentation - Results:** The larger the window size the more computation time is the RDFization of raw data. As an example the RDFization process of a 10 minutes window of a SIRI file (i.e., 30 snapshots described by 9565 lines) was achieved in 6720.4 ms, which gives a processing time of 0.70 ms/line. This transformation generated 97297 RDF triples, which gives an average throughput of 14477.86 triples/sec. We also note that the proportion of computation time vs. detection and diagnosis evolves with respect to the window size. For instance the RDFization process represents 82% of the overall process for a window size of 60 while its represents "only" 63% for a size of 3. The computation time of the detection and diagnosis process represents, on average, respectively 14% and 5% of the overall process. We also note that more heterogeneity in the sources for diagnosis (C_1 vs. C_2) the more time consuming is the RDFization. The quasi real-time aspect of our approach is preserved as 19.5 seconds is required in the worst case (i.e., < 20 seconds update of SIRI data).

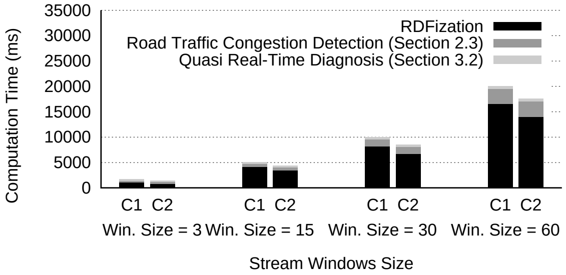

Fig. 8. Computation Time of the Overall Approach.

• **Semantic Matching Experimentation - Context:** Fig.9 illustrates the computation time of the different DL reasoning used for computing the quasi real-time diagnosis i.e., DL subsumption (for evaluating Exact, PlugIn, Subsume, Intersection and Disjoint-based comparison of descriptions) and concept abduction (for reporting back the diagnosis approximation). The window of exploration of the dynamic knowledge is experimented from a range of 1 min (i.e., 3 snapshots) to 120 min (i.e., 360 snapshots) over a busy week-end (for maximizing the number of events). The complete context of Section 4.1 has been considered in this experiment i.e., historical data of 217 days for road works, city events, weather conditions and weather information. Only historical data which fit the same time are considered.

• **Semantic Matching Experimentation - Results:** Our approach guarantees to obtain diagnosis report in suitable range of computation time: from 0.3 to 3.5 seconds for a window of exploration of respectively 1 and 120 minutes. Subsumption is the more time consuming reasoning since a large number of subsumption tests have to be performed between various real-time and historical events, road works, road conditions, weather information. Fig.9 shows a steep growth, mainly due to the increasing number of potential matching tests that required to be evaluated when extending the window size. Its computation becomes really problematic if the diagnosis is estimated on a larger windows e.g., 9.6 seconds is required for a window of 540 snapshots. However diagnosis used to be computed through an analysis up to 90 snapshots. The abduction computation does not vary significantly (on average 0.1 second) mainly because only one diagnosis report is elaborated, independently of the size of the stream window.

4.4 Lessons Learned

During the transformation of raw data into semantic description, we were facing the challenge of discovering the appropriate vocabulary, with the appropriate expressivity. We mainly used LOD vocabularies for linkage and integration with external source of data. However, some cases (i.e., Dublin Bus, events and road works data) required more specific and fine-grained descriptions with higher expressivity for matching and reasoning purposes. Towards this issue we carefully developed our own terminologies, aligned with the schema of raw data, for reusability and reasoning.

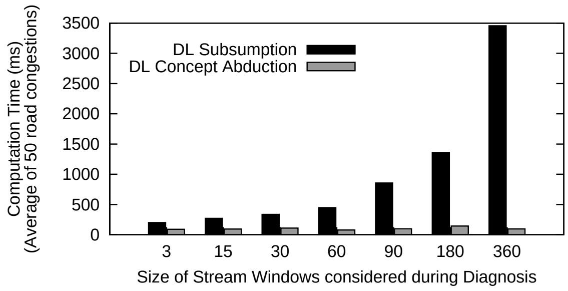

Fig. 9. Computation Time for Semantic Matching.

The on-the-fly integration/linkage of new sources of causes (Table 1), exposed as raw data, with our existing RDFized data is not straightforward even with existing tools. Most relevant linkages had to be done manually during the first integration e.g., any road work is a sort of event in our context but none of existing tool has infer such a link.

We enriched exiting data sets from Dublin City with OWL $\mathcal{EL}^{++}$ based description not only for computing matching between events, road works (among others), but also to report the impact of approximating diagnosis conditions (through concept abduction). It is obvious that the computation performance of our approach would have been strongly altered when considering much more expressive semantics such as OWL 2 Full or DL. Indeed the quasi real-time dimension cannot be met in OWL 2 Full or DL due to the complexity of subsumption and abduction. However considering OWL 2 Full or DL could have triggered more causes for road congestions, and improved the diagnosis precision. It would be also interesting to evaluate the impact of using a subset of $\mathcal{EL}^{++}$ on the computation performance and the diagnosis results. In other words, which *expressiveness does fit the better for this application* is still an open challenge. Further experiments are required to provide the most appropriate context and trade-off complexity/expressivity.

The transformation of SIRI data to RDF is a crucial part and the most time consuming part of the overall approach (Fig.8). Using pure Jena classes[17] to add triples requires 15 minutes to RDFize a 90 minutes window of SIRI data. In order to meet our quasi real-time constraint our platform provides a customized Java *InputStream* of RDF triples, that are generated in a buffered way by reading one or more CSV files according to the requested time window. We feed our customized *InputStream* to the class *TDBLoader*, which allows bulk loading into a TDB dataset. This approach requires less than 60 seconds to RDFize a 90 minutes window of SIRI data

[17] http://jena.apache.org/

5 Related Work

There are many approaches in traffic controls where domain experts are in charge of understanding effects of specific and targeted events on road conditions in order to take corrective actions. However the automated and real-time explanation of traffic congestions has not been tackled, making road traffic difficult to be efficiently managed.

Diagnosis, or the process of identifying the nature and cause of an anomaly (aka. conflict) has been largely studied by the Semantic Web community, but mainly in the context of an ontology. Existing works [16] applied and extended axioms pinpointing to derive how changes in an ontology may result in conflicts in the knowledge base. Following [17], the task of *diagnosis* consists in retrieving the causes of an anomaly (e.g., road congestions) by interpreting external sources of causes. We extended it by considering historical information to capture potential diagnosis. We make use of existing semantic matching techniques to approximate diagnosis in an open-world scenario i.e., new sources of causes of anomalies (e.g., road works). In addition concept abduction is considered to report back the information we gained or lost during diagnosis.

There are no existing approaches that integrate semantics and diagnosis techniques. However, its integration is needed since existing approaches cannot handle new events and observations as we consider in this work. They all assume a closed world scenario where the set of possible causes that could explain the effects is well defined and where cause-effect relationships can (at least with unlimited computational resources) be established. The closest diagnosis works to our approach are the ones that tackle the complexity problem of diagnosis approaches [18] by precomputing diagnosis results for *some* anomalies. If other anomalies are detected some machine learning methods are used to estimate the diagnosis result in these cases [19]. However, this estimation consists only of a numeric value rather than an expressive (semantic) explanation as in our case. Furthermore these approaches consider only the problem of mapping anomalies to well defined sets of possible causes rather than to new causes as in our case.

Semantic web technologies and machine learning techniques have been coupled by [20] for (i) road traffic prediction and (ii) trip planning in Milano City. Our work goes further by explaining traffic congestions by revisiting AI diagnosis. Our work required a higher level of expressivity for interpreting diagnosis results in open-world scenarios.

[21] present a framework for publishing, RDFizing and linking transport data on the Web. Contrary to our work, they do not consider the stream dimension of transportation data, and no quasi real-time RDFization is presented. We targeted different applications where ours required more expressive and specific ontologies.

6 Conclusion and Future Work

Diagnosis, *or the method to connect causes to its effects*, is an important reasoning task for obtaining insight on cities, its road traffic and reaching the concept of sustainable and smarter cities that is envisioned nowadays. This work focused on diagnosing road traffic congestions in the real-world context of Dublin City where static and stream data of its road traffic domain has been exploited. Our approach coupled pure AI diagnosis approaches with semantic web technologies for accurate and quasi real-time diagnosing in an open-world context of heterogeneous and large data. The approach has shown high performance and applicability in the context of real and live data from Dublin City. In addition we raised some challenges we met during the implementation of the prototype.

We currently study the integration of our approach with IBM DB2 RDF and expect to serve real-time semantic streams by using IBM InfoSphere Streams. In future work, we will further evaluate the impact of the number of other data sources (e.g., real-time CCTV monitoring of Dublin City) on precision and scalability of the diagnosis approach. We also expect using citizen sensing (e.g., twitter traffic data) to validate diagnosis results. Finally, we will work on a model for predicting road traffic congestions by coupling semantic web technologies and machine learning approaches.

References

1. Biem, A., Bouillet, E., Feng, H., Ranganathan, A., Riabov, A., Verscheure, O., Koutsopoulos, H.N., Moran, C.: Ibm infosphere streams for scalable, real-time, intelligent transportation services. In: SIGMOD. (2010) 1093–1104
2. Luo, C., Thakkar, H., Wang, H., Zaniolo, C.: A native extension of sql for mining data streams. In: SIGMOD Conference. (2005) 873–875
3. Babu, S., Widom, J.: Continuous queries over data streams. SIGMOD Record **30**(3) (2001) 109–120
4. Haase, C., Lutz, C.: Complexity of subsumption in the [escr][lscr] family of description logics: Acyclic and cyclic tboxes. In: ECAI. (2008) 25–29
5. Baader, F., Brandt, S., Lutz, C.: Pushing the el envelope. In: IJCAI. (2005) 364–369
6. Horrocks, I., Sattler, U.: Ontology reasoning in the shoq(d) description logic. In: IJCAI. (2001) 199–204
7. Ren, Y., Pan, J.Z.: Optimising ontology stream reasoning with truth maintenance system. In: CIKM. (2011) 831–836
8. Huang, Z., Stuckenschmidt, H.: Reasoning with multi-version ontologies: A temporal logic approach. In: ISWC. (2005) 398–412
9. Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., Teneketzis, D.: Failure diagnosis using discrete event models. IEEE Trans. on Cont. Sys. Tech. **4**(2) (1996) 105–124
10. Sinnott, R.W.: Virtues of the haversine. Sky and Telescope **68**(2) (1984) 159–197
11. Schumann, A., Pencolé, Y., Thiébaux, S.: Symbolic models for diagnosing discrete-event systems. In: ECAI. (2004) 1085–1086
12. Paolucci, M., Kawamura, T., Payne, T., Sycara, K.: Semantic matching of web services capabilities. In: ISWC. (2002) 333–347
13. Li, L., Horrocks, I.: A software framework for matchmaking based on semantic web technology. In: WWW. (2003) 331–339
14. Noia, T.D., Sciascio, E.D., Donini, F.M., Mongiello, M.: Abductive matchmaking using DLs. In: IJCAI. (2003) 337–342
15. Scherp, A., Franz, T., Saathoff, C., Staab, S.: F–a model of events based on the foundational ontology dolce+dns ultralight. In: K-CAP. (2009) 137–144
16. Parsia, B., Sirin, E., Kalyanpur, A.: Debugging owl ontologies. In: WWW. (2005) 633–640
17. Lécué, F.: Diagnosing changes in an ontology stream: A dl reasoning approach. In: AAAI. (2012)
18. de Kleer, J., Mackworth, A.K., Reiter, R.: Characterizing diagnoses and systems. Artificial Intelligence **56**(2–3) (1992) 197 – 222
19. Keren, B., Kalech, M., Rokach, L.: Model-based diagnosis with multi-label classification. In: 22nd International Workshop on Principles of Diagnosis (DX-11). (2011)
20. Valle, E.D., Celino, I., Dell'Aglio, D., Grothmann, R., Steinke, F., Tresp, V.: Semantic traffic-aware routing using the larkc platform. IEEE Internet Computing **15**(6) (2011) 15–23
21. Plu, J., Scharffe, F.: Publishing and linking transport data on the web. International Workshop On Open Data **abs/1205.1645** (2012)