An Expert System for Water Quality Monitoring Based on Ontology

Edmond Jajaga[1(✉)], Lule Ahmedi[2], and Figene Ahmedi[3]

[1] Department of Computer Science, South East European University, Ilindenska n. 335 1200, Tetovë, Macedonia
e.jajaga@seeu.edu.mk

[2] Department of Computer Engineering, University of Prishtina, Kodra e diellit pn 10000, Prishtinë, Kosova
lule.ahmedi@uni-pr.edu

[3] Department of Hydro-Technic, University of Prishtina, Kodra e diellit pn 10000, Prishtinë, Kosova
figene.ahmedi@uni-pr.edu

Abstract. Semantic technologies have proved to be a suitable foundation for integrating Big Data applications. Wireless Sensor Networks (WSNs) represent a common domain which knowledge bases are naturally modeled through ontologies. In our previous works we have built domain ontology of WSN for water quality monitoring. The SSN ontology was extended to meet the requirements for classifying water bodies into appropriate statuses based on different regulation authorities. In this paper we extend this ontology with a module for identifying the possible sources of pollution. To infer new implicit knowledge from the knowledge bases different rule systems have been layered over ontologies by state-of-the-art WSN systems. A production rules system was developed to demonstrate how our ontology can be used to enable water quality monitoring. The paper presents an example of system validation with simulated data, but it is developed for use within the InWaterSense project with real data. It demonstrates how Biochemical Oxygen Demand observations are classified based on Water Framework Directive regulation standard and provide its eventual sources of pollution. The system features and challenges are discussed by also suggesting the potential directions of Semantic Web rule layer developments for reasoning with stream data.

Keywords: Expert system · Semantic Web · Ontology · Metadata · SSN · Big Data · Stream data

1 Introduction

Social networks, logging systems, sensor networks etc. are delivering huge amount of continuous flow of data also known as stream data. More data are produced more machine intelligence is required to deal with them. Streaming technologies like Complex Event Processing (CEP), Data Stream Management Systems, and Stream Reasoning (SR) are supporting Big Data applications development. According to a survey

© Springer International Publishing Switzerland 2015
E. Garoufallou et al. (Eds.): MTSR 2015, CCIS 544, pp. 89–100, 2015.
DOI: 10.1007/978-3-319-24129-6_8

conducted by Gartner Inc. 22% of the 218 respondents with active or planned big data initiatives said they were using stream or CEP technologies or had plans to do so [7]. In particular, SR provides a high impact area for developing powerful applications for analyzing stream data. State-of-the-art stream data knowledge bases are merely modeled through ontologies. Ontologies, in particular OWL ontologies, are mainly modeled in Description Logic (DL). Reasoning in ontological terms is not enough to express real-world application scenarios. For example, deriving new and implicit knowledge from ontologies is efficiently done through rule-based reasoning. However, layering Semantic Web rule-based DL systems, such as SWRL, over DL ontologies lacks the expressivity to handle some reasoning tasks, especially for the domain of SR e.g. finding average values [1]. A lot of research has been taken by the SR community to address data management and query processing on streaming data [4], while little efforts have been taken to address the stream reasoning inference problems [14]. In absence of a proper Semantic Web rule system different ones have been layered over stream data ontology bases. In our previous works in [1, 2], we have discussed about pros and cons for approaching hybrid and homogeny solutions. Mainly, the reasons for passing to hybrid solutions include non-monotonicity issues and solving complex reasoning tasks.

InWaterSense[1] is a R&D project for developing intelligent WSNs for WQM which objectives include:

- Build a Wireless Sensor Networks (WSN) infrastructure in the river Sitnica for monitoring water quality with the aim of providing a best practice scenario for expanding it to other surface water resources as well in the Republic of Kosovo.
- Monitor water quality in the river Sitnica supported by the WSN in order to make the quality data available to the community and the decision makers for determining the current health of the river.
- Transform the existing WSN for WQM into an intelligent platform to operate almost autonomously, and support more functionality as envisioned by the future Internet and intelligent systems.

In line with our project objectives, especially the later one, we have built INWATERSENSE (INWS) ontology framework [2], a SSN[2]-based ontology for modeling WQM domain. An extension of this ontology is developed for enabling identification of the potential polluter. Moreover, an expert system, using the Java Expert System Shell (Jess) [11], was developed to reason over INWS ontology. Jess is a rule engine and scripting environment written in Java. The contribution illustrates the main characteristics of an expert system for WQM. Namely, it classifies water bodies based on observed water quality values and investigates eventual sources of water quality degradation. We discuss the features and challenges of this system while also addressing its potential improvements. Since we plan in the future to build a pure Semantic Web framework for WQM, we also discuss the main challenges expected for building such system. The Jess expert system described in this paper will then be compared with this system.

---

[1] http://inwatersense.uni-pr.edu/

[2] Semantic Sensor Network Ontology, http://purl.oclc.org/NET/ssnx/ssn

The paper is organized as follows. We begin in the following section with description of INWS ontology model and the requirements for rule-based stream data reasoning. Section 3 describes the conceptual architecture of our SR framework for WQM. The expert system implementation is described in Section 4, while its challenges and discussions together with the pure Semantic Web approach are presented in Section 5. The paper ends with the concluding notes and future plans.

2 Background

The INWS ontology framework [2] models the WSN for WQM into three modules: core[3], regulations[4] and pollutants[5]. The core ontology extends the SSN ontology to meet the requirements for a WSN for WQM. It models WSN infrastructure entities, observations, observation time and location and water quality parameters. The regulations ontology models classification of water bodies based on different regulation authorities such as Water Framework Directive (WFD) [17]. And finally, the pollutants ontology models the entities for investigating sources of pollution.

A typical scenario for WQM in a WSN platform is as below:

Scenario 1. Water quality sensor probes are deployed in different measurement sites of a river. A sensor probe emits water quality values. We want to (1) classify the water body into the appropriate status according to WFD regulations and (2) identify the possible polluter if the values are below the allowed standard.

In order to handle the requirements of this scenario, a SR system should support reasoning over both streaming information and background data [19]. In particular, to enable efficient rule-based reasoning over stream data we address some specific requirements about this property which are already mentioned in state-of-the-art systems e. g. StreamRule [25]. Namely, a SR rule systems need to support a combination of reasoning features like: closed-world, non-monotonic, incremental and time-aware reasoning.

Since the Web is open and accessible by everyone, Semantic Web recommended standards (OWL and SWRL) manage knowledge bases in terms of open world assumption (OWA). In OWA, if some knowledge is missing it is classified as undefined as opposed to the closed-world assumption (CWA) which classifies the missing information as false. In the Web, addition of new information does not change any previously asserted information which is known as monotonic reasoning. This is not the case with non-monotonic reasoning during which addition of new information implies eventual modifications in the knowledge base. In SR application domains, OWL and SWRL's OWA and monotonic reasoning do not offer the desired expressivity level. For example, modifying the river pollution status is not allowed through SWRL rules. Following the SWRL's monotonic nature a river instance firstly asserted as "clean" cannot be later modified as "polluted".

[3] http://inwatersense.uni-pr.edu/ontologies/inws-core.owl
[4] http://inwatersense.uni-pr.edu/ontologies/inws-regulations.owl
[5] http://inwatersense.uni-pr.edu/ontologies/inws-pollutants.owl

Inferring new implicit data from stream data will result in multiple CRUD operations, which in SR is known as incremental reasoning. In our case study, new coming sensor observation data need to be consumed quickly and together with previously inferred data will serve as for inferring new implicit data.

SR systems should also include time-annotated data i.e. the time model, and like CEP should offer explicit operators for capturing temporal patterns over streaming information [19]. The INWS ontology layer implements the time model through OWL Time ontology[6]. Supporting temporal operators (serial, sequence, etc.) means the system can express the following example rule: *Enhanced phosphorus levels in surface waters (that contain adequate nitrogen) can stimulate excessive algal growth* [18]. If before excessive algal growth, enhanced phosphorus level has been observed then more probably the change of phosphorus levels has caused the algal growth. Thus, a sequence of these events needs to be tracked to detect the occurrence of this complex event.

Moreover, in order to enable reasoning in terms of time and quantity intervals of continuous and possibly infinite streams the SR notion of windows need to be adopted for rules [13]. For example, a rule to assert which sensors provided measurements that are above allowed average threshold the last 30 minutes sliding the window every 5 minutes, will be easily expressible with the help of the window concept. This has raised the need for a specific kind of rules in SR, namely continuous rules. Rather than evaluating rules against almost static ABox knowledge base as in traditional Semantic Web rule systems, continuous rule-based reasoning must run over dynamic stream data instead. With the set of new-coming data streams new logical decisions will arise: new information need to be published on the knowledge base or a fact modification/retraction need to be performed.

3 System Architecture

As depicted in Fig. 1, our system's architecture consists of three layers: *data, INWS ontology* and *rules layer*. The RDF data (up left) and RDF streams (up right) constitute the *data layer* (grey track). Arrows describe data flow direction. Domain specific ABox knowledge which does not change or changes "slowly" is formulated in the form of RDF data e.g. river names. RDF streams are defined as a sequence of RDF triples that are continuously produced and annotated with a timestamp [9]. Water quality measured values, annotated as RDF streams, will continuously populate the core ontology. In particular, a single RDF stream will hold information of observed water quality value, timestamp and location. The middle part of Fig. 1 represents the *INWS ontology* (green track) described in the previous section. The *rule layer* (yellow track) consists of common rules (bottom left) and continuous rules (bottom right). In the previous section we mentioned the concept of continuous rules which should infer new implicit knowledge from RDF streams.

[6] http://www.w3.org/TR/owl-time/

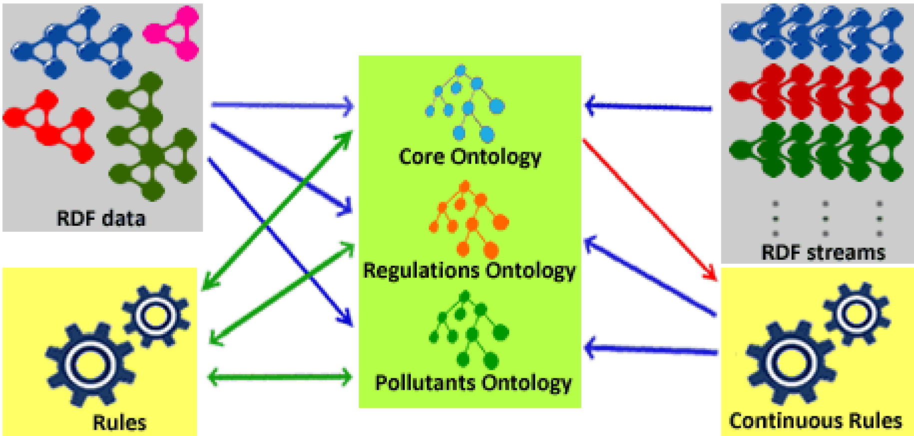

Fig. 1. InWaterSense conceptual framework: *data layer* (grey track), *ontology layer* (green track) and *rules layer* (yellow track)

4 Implementation

We decided to use Jess as a platform for implementing our system of reasoning over the INWS ontology framework. As a production rule system, Jess supports closed-world and non-monotonic reasoning. Moreover, it has a tight integration with Java through Jess's Java API and Protégé through JessTab[7] plugin. JessTab is a plug-in for the Protégé[8] ontology editor and knowledge-engineering framework that allows one to use Jess and Protégé together. The system is validated with simulated data, but it is developed for use within the InWaterSense project with real data.

The Jess implemented architecture of our system and its related components for reasoning over the INWS ontology are presented in Fig. 2. Namely, input data in their available format, say SQL, are transformed into RDF streams using D2RQ[9] tool. SWOOP [12] is used to load the D2RQ generated RDF data and produce the abbreviated RDF/XML syntax for object property instances to be readable by Protégé [2]. RDF data streams are next imported into the core ontology. The set of rules for water quality classification based on WFD regulations are defined and may run against the knowledge base. Moreover, a set of rules for investigating sources of pollution by observing if eventual critical events appear are defined and may be activated. A simple user interface was developed using Java Swing[10], which offers a user to monitor water quality based on the WFD regulations and to eventually find the possible sources of pollution.

---

[7] http://www.jessrules.com/jesswiki/view?JessTab
[8] Protégé ontology editor, http://protege.stanford.edu/
[9] D2RQ Accessing Relational Databases as Virtual RDF Graphs, http://d2rq.org/
[10] http://openjdk.java.net/groups/swing/

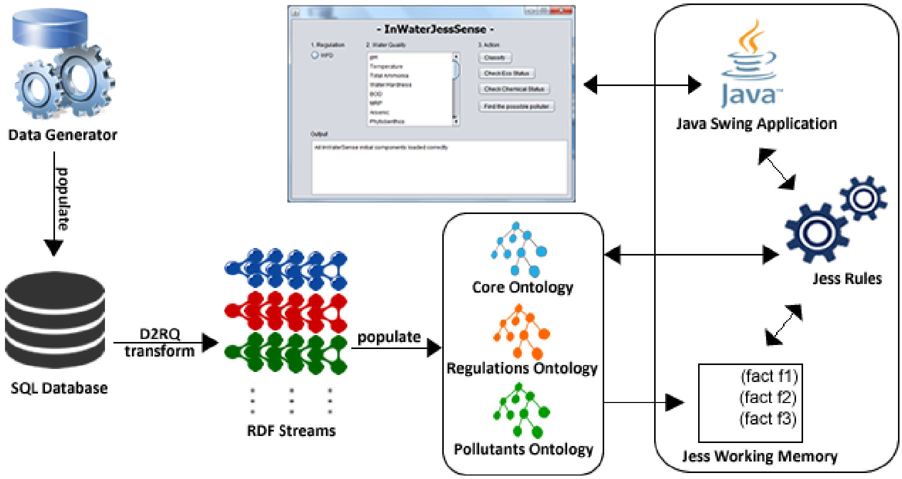

Fig. 2. Jess implemented architecture for WQM

4.1 The Pollutants Ontology

The INWS pollutants ontology was designed based on examples of sources of pollution and the potential pollutant discharges which could arise described in [18]. Two classes are added: `PollutionSources`, describing the sources of pollution e.g. urban storm water discharges, and `Pollutants`, representing contaminants present in the environment or which might enter the environment which, due to its properties or amount or concentration, causes harm e.g. heavy metals. A property `potentialPollutant` links individuals of `PollutionSources` and `Pollutants` (based on the Table on page 3 in [18]). `PollutionSources` class is also linked with a string through two properties: `pollutionSourceName`, representing the name of the pollution source, and `pollutionType`, representing the type of the pollution source which can be point, diffuse or both of them. Moreover, a property `hasSourcesOfPollution` was added to relate the rivers with the sources of pollution.

4.2 Implementation of the Scenario

To implement the *Scenario 1* using our system interface, as depicted in Fig. 3, one should select the regulation authority i.e. WFD, select the water quality parameters which are to be monitored and press the button “Classify”. The JTextArea below the “Output” label serves for printing rules messages.

The system offers multiple selections of water quality parameters. A simple rule is fired at application startup to set up the observations interval beginning time from the earliest time of observations streams and end time from the latest one. For brevity and clarity, we will demonstrate Biochemical Oxygen Demand (*BOD5*) observations WFD classification. According to WFD regulations: *if BOD_5 observations’ average value is between 1.3 and 1.5 mg O_2/l then river belongs to “Good” status of oxygen* *condition, if the average is below 1.3 then river belongs to “High” status, else the river belongs to “Moderate” status*. Expressing this rationale with Jess rules was done through a number of rules. Namely, a rule of primer priority creates auxiliary Jess facts holding BOD_5 measurement values coming from the RDF streams. We should have used observation values directly from the ontology mappings but the Jess rule which calculates the average value constrains the usage of Jess facts.

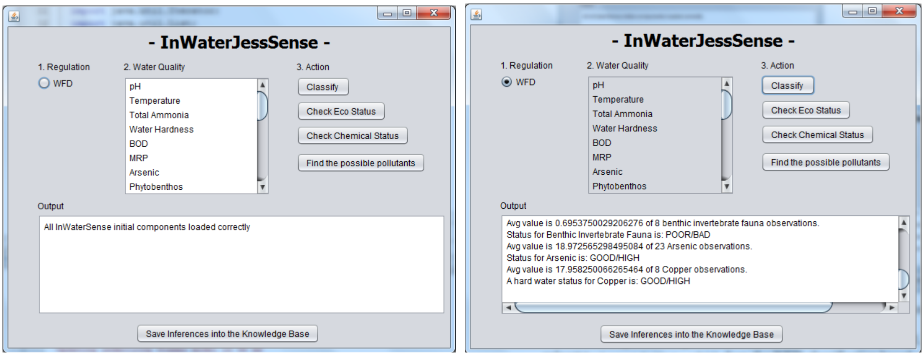

Fig. 3. The Jess system interface: initial view (left) and after WFD classification view (right)

After finding the average value it is asserted as a fact into the WM. Finally, another rule `WFDclassifyWaterBOD` does the WFD classification based on the previously asserted average value. This rule is illustrates below:

```
1 (defrule WFDclassifyWaterBOD
2 (BODaverage (v ?x)) (CurrentInterval (v ?i)) =>
3 (if (and (< ?x 1.5) (> ?x 1.3)) then (and
4 (printout t "Status for BOD is: GOOD" crlf)
5 (make-instance (str-cat "GoodBODStatus" ?*r*) of http://.../inws-
    regulations.owl#GoodBODMeasurement map)
6 (make-instance (str-cat "ObservationInstantBOD" ?*r*) of
    http://.../inws-regulations.owl#ObservationInstant map)
7 (slot-insert$ (str-cat "ObservationInstantBOD" ?*r*)
8 http://www.w3.org/2006/time#inXSDDateTime 1 ((new Date) toString))
9 (slot-insert$ (str-cat "ObservationInstantBOD" ?*r*)
10  http://.../inws-regulations.owl#hasStatus 1
11  (str-cat "http://.../inws-core.owl#GoodBODStatus" ?*r*))
12 (slot-insert$ (str-cat "http://.../inws-core.owl#" ?i)
13  http://.../inws-regulations.owl#hasStatus 1
14  (str-cat "http://.../inws-core.owl#GoodBODStatus" ?*r*))))
15 (if (< ?x 1.3) then <HIGH status classification code here>)
16 (if (> ?x 1.5) then <MODERATE status classification code here>))
```

Code in Line 1 serves for declaring a rule definition and its name. Line 2 represent the left hand side of the rule while lines 3-16 the right hand side of the rule. The previously calculated average value is assigned to variable `?x` while the current interval of observations present in the WM is assigned to `?i` (Line 2). If `?x` is between 1.5 and 1.3 begin assertions for good status (Line 4-14). Namely, a message is printed out (Line 4); a new instance of regulations ontology class `GoodBODMeasurement` is created (Line 5) (`?*r*` is a global variable holding random integer numbers); a new instance of `ObservationInstant` class is created (Line 6) associated with current date and time through `inXSDateTime` property (Line 7-8). This instance is also related with the instance created in Line 5 through `hasStatus` property (Line 9-11). Current interval instance (Line 12) is associated with the newly asserted status instance (Line 13-14). The same steps presented in line 4-14 are performed for the high and moderate status, which are omitted for brevity (Line 15-16).

The second part of *Scenario 1* is encoded through a couple of rules. The first one detects newly asserted instances of moderate status i.e. instances of `ModerateBOD-Measurement` class. If there is at least one instance the second rule will fire and find BOD_5 sources of pollution discharging in the river body. An example of BOD_5 observations status is illustrated in Fig. 4. BOD_5 sources of pollution are also listed after the user has clicked the "Find possible pollutants" button.

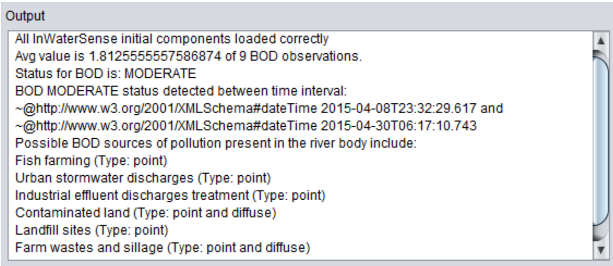

Fig. 4. *Scenario 1* example output for BOD_5 observations WFD classification and sources of pollution

5 Challenges and Discussion

In this section will be discussed the features of the Jess system and the challenges to be addressed for its further improvements. Meanwhile, potential future directions for building a pure Semantic Web rule system, such as SWRL, for WQM also take place in the discussion. This system is planned to support time-aware, closed-world, non-monotonic and incremental reasoning to enable stream data reasoning.

5.1 Continuous Rules

The Jess system effectively identifies water quality status for the set of input RDF streams. Upcoming RDF streams are collected into another set of streams which in turn are imported into INWS ontology for rule-based reasoning. As per future works we plan to automate this process. Namely, RDF streams coming from SQL through D2RQ translation will continually populate the ontology and be automatically mapped into Jess's WM. Meanwhile, with the time passing old facts will be discarded from the WM and be deployed into the knowledge base for future reasoning. If a class is mapped through JessTab command `mapclass` then it will place all its instances into the WM. This is not practical with stream data as data flow is massive and rules will consider a specific set (time or quantity constrained window) of RDF streams. A workaround solution would be to create Jess facts out of window's selected Protégé instances. But this way the WM will hold Protégé instances and their one or many Jess facts copies. Moreover, instead of producing a query output results like in C-SPARQL, the continuous firing of rules will continually modify the knowledge base i. e. do incremental reasoning. This is efficiently done through JessTab functions for manipulation of ontology classes and instances. However, using inferred knowledge between observation RDF streams sets is planned for future system improvements.

5.2 Logic Foundation

The core issue for building a pure Semantic Web system for stream data from which follow the respective expressivity constrains is the system's underlying logic foundation. Production rules and LP implementations has shown great success in the domain of SR. Different Semantic Web applications fall into different logic domains and possibly in a mixture of them [6]. The authors of [6] conclude that the Description Logic Programs (DLP) fragment should offer extension for both LP and DL. In the area of SR, DL reasoning fulfills the requirements for modeling the knowledge bases. When it comes to rule-based reasoning DL's OWA limits the expressivity power for even simple reasoning tasks (e.g. counting class instances). Since LP adopts CWA approach together with non-monotonicity an LP extension of DLP, would be ideal for the WQM case study and stream data in general.

5.3 Forward/Backward-Chaining and Rete Algorithm

Inferring new knowledge in rule systems can be done in two methods: deriving conclusions from facts, known as forward-chaining or starting from conclusion (goal) by matching facts also known as backward-chaining.

In production rule systems, matching rules with relevant facts is efficiently done with the Rete algorithm [9]. Executing rules through Rete algorithm means all relevant facts must be loaded into the WM [3]. Considering the massive flow of stream data the WM will become overwhelmed. Adding here the amount of the inferred facts, the memory will become exhausted. With the introduction of the continuous rules this issue will be resolved by capturing only snapshots (time-based or count-based windows) of streams and thus facts will enter and leave WM as needed.

In SR applications facts are changing very often, while rules change "slowly". Newly inserted facts in WM will cause rule firing. This intuitively indicates the forward-chaining nature of stream data applications. Rete algorithm natively adopts forward-chaining approach. However, the traditional Rete algorithm does not support aggregation of values in time-based windows [20]. Authors in [20] present a CEP system which extends Rete algorithm for supporting time-aware reasoning by leveraging the time-based windows and enabling calculation of complex events e.g.

finding average value. They have added a time enabled beta-node to restrict event detection to a certain time-frame. On the other side, Semantic Streams [21], prove that backward-chaining can also be enabled on stream data even though its slight modification has been needed to produce the legal flow of data.

5.4 Hybrid and Homogeny Stream Data Approaches

State-of-the-art rule-based systems for reasoning over stream data mainly fall into two broad categories: hybrid and pure Semantic Web approaches [1].

Hybrid approaches layer different rule systems over ontologies like: production rules, CEP, LP, Answer Set Programming etc. In our previous work [1] we described in more detail about each approach and their pros and cons. In general, hybrid ones have achieved the desired system behavior by translating the ontology into the corresponding formalisms of the overlaying rule system. A drawback of this translation is that a possible loss of information may occur. For example, translating complex subclass statements consisting of disjunction of classes or expressed with existential quantification are not possible into Plausible Logic [5]. Moreover, when adding a rule a possible side-effect may occur. For example, in production rule systems adding a rule may require extra work because of the algorithm used for executing the rules as depicted in [3]. This makes it harder for domain experts to write rules without IT support. In some cases (as shown in [3]) development layers are conflate to each other making rules maintenance more laborious. SWRL on the other side is declarative rule language not bound to any particular execution algorithm [3]. However, equipping SWRL with non-monotonic reasoning means the order of rules should be taken into account [24]. StreamRule demonstrates how non-monotonic, incremental and time-aware reasoning can be integrated into a unique platform for stream data reasoning. However, the continuous rule feature is implemented through separate steps. Namely, stream filtering and aggregation is done through a stream query processor such as CQELS [26] while OClingo [27] is used to enable non-monotonic reasoning.

Pure Semantic Web approaches like [22] and [23] do not make any distinction between stream and random data and lack implementation. These approaches prove that SWRL can be used to infer new and approximate knowledge in stream data domains. The work presented in [16] describes a Rete-based approach of RIF rules for producing data in a continuous manner. Although supporting time-aware and incremental reasoning, the approach does not deal with non-monotonic and closed-world reasoning. Rscale [8] is another industrially-approved reasoning system which leverages OWL 2 RL language profile to infer new knowledge. It enables incremental reasoning, non-monotonic and closed-world reasoning through translation of facts and rules into SQL tables and queries respectively. However, it does not support time-aware reasoning, and as a non-Semantic Web approach follows the hybrid approach disadvantages. JNOMO [24] shows how SWRL can be extended to embrace non-monotonicity and CWA. However, inclusion of temporal reasoning is envisioned as per future works.

6 Conclusion and Future Work

The main contributions of this paper include an extension of INWS ontology with metadata descriptions for water quality pollution sources and an expert system that uses INWS ontology to enable WQM. The system efficiently classifies water bodies based on WFD standards encoded into Jess rules running over the set of observations RDF streams. Moreover, a set of Jess rules are used to detect the eventual sources of pollution. However, the notion of windows needs to be adopted for Jess rules to enable continuous rules feature. The system's features and challenges were also discussed as lessons learned for future plans of building Semantic Web homogeny solution for reasoning over stream data. Forward-chaining reasoning method is a natural approach for stream data while an LP extension to DLP was also identified as a suitable underlying logic for the rule system reasoning over stream data. Our future works also include the evaluation of the expert system described in this paper and comparing it with the pure Semantic Web system.

Acknowledgements. The work described in this paper was supported by "InWaterSense: Intelligent Wireless Sensor Networks for Monitoring Surface Water Quality", an EU funded project managed by European Union Office in Kosovo, implemented by University of Prishtina.

References

1. Jajaga, E., Ahmedi, L., Abazi-Bexheti, L.: Semantic Web trends on reasoning over sensor data. In: 8th South East European Doctoral Student Conference, Greece (2013)
2. Ahmedi, L., Jajaga, E., Ahmedi, F.: An ontology framework for water quality management. In: Corcho, Ó., Henson, C.A., Barnaghi, P.M. (eds.) SSN@ISWC, Sydney, pp. 35–50 (2013)
3. MacLarty, I., Langevine, L., Bossche, M.V., Ross, P.: Using SWRL for Rule Driven Applications. Technical report (2009)
4. Unel, G., Roman, D.: Stream reasoning: a survey and further research directions. In: Andreasen, T., Yager, R.R., Bulskov, H., Christiansen, H., Larsen, H.L. (eds.) FQAS 2009. LNCS, vol. 5822, pp. 653–662. Springer, Heidelberg (2009)
5. Groza, A., Letia, I.A.: Plausible description logic programs for stream reasoning. Future Internet **4**, 865–881 (2001)
6. de Bruijn, J., Polleres, A., Lara, R., Fensel, D.: OWL DL vs. OWL flight: conceptual modeling and reasoning for the semantic web. In: Fourteenth International World Wide Web Conference, pp. 623–632. ACM, Chiba (2005)
7. Report Highlight for Survey Analysis: Big Data Investment Grows but Deployments Remain Scarce in 2014, http://www.gartner.com/newsroom/id/2848718
8. Liebig, T., Opitz, M.: Reasoning over dynamic data in expressive knowledge bases with rscale. In: The 10th International Semantic Web Conference, Bonn, Germany (2011)
9. Forgy, C.L.: Rete: A fast algorithm for the many pattern/many object pattern match problem. Artificial Intelligence **19**(1), 17–37 (1982)
10. Barbieri, D.F.: C-SPARQL: SPARQL for continuous querying. In: Proceedings of the 18th International World Wide Web Conf. (WWW 2009), pp. 1061–1062 (2009)

11. Hill, E.F.: Jess in action: java rule-based systems. Manning Publications Co., Greenwich (2003)
12. Horrocks, I., Patel-Schneider, P.F., Boley, H., Tabet, S., Grosof, B., Dean, M.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML (2004)
13. Stuckenschmidt, H., Ceri, S., Della Valle, E., van Harmelen, F.: Towards expressive stream reasoning. In: Proceedings of the Dagstuhl Seminar on Semantic Aspects of Sensor Networks (2010)
14. Della Valle, E., Schlobach, S., Krötzsch, M., Bozzon, A., Ceri, S., Horrocks, I.: Order matters! harnessing a world of orderings for reasoning over massive data. Semantic Web Journal **4**(2), 219–231 (2013)
15. Sheth, A., Henson, C., Sahoo, S.S.: Semantic sensor web. IEEE Internet Computing **12**(4), 78–83 (2008)
16. Albeladi, R., Martinez, K., Gibbins, N.: Incremental rule-based reasoning over RDF streams: an expression of interest. In: RDF Stream Processing Workshop at the 12th Extended Semantic Web Conference, Portoroz, Slovenia (2015)
17. Directive 2000/60/EC of the European Parliament and of the Council of Europe of 23 October 2000 establishing a framework for Community action in the Field of water quality O.J. L327/1 (2000)
18. Sources of Pollution, Foundation for Water Research, Information Note FWR-WFD16 (2005)
19. Margara, A., Urbani, J., van Harmelen, F., Bal, H.: Streaming the web: reasoning over dynamic data. Web Semantics: Science, Services and Agents on the World Wide Web **25**, 24–44 (2014)
20. Walzer, K., Groch, M., Breddin, T.: Time to the rescue - supporting temporal reasoning in the rete algorithm for complex event processing. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2008. LNCS, vol. 5181, pp. 635–642. Springer, Heidelberg (2008)
21. Whitehouse, K., Zhao, F., Liu, J.: Semantic streams: a framework for composable semantic interpretation of sensor data. In: Römer, K., Karl, H., Mattern, F. (eds.) EWSN 2006. LNCS, vol. 3868, pp. 5–20. Springer, Heidelberg (2006)
22. Wei, W., Barnaghi, P.: Semantic annotation and reasoning for sensor data. In: Barnaghi, P., Moessner, K., Presser, M., Meissner, S. (eds.) EuroSSC 2009. LNCS, vol. 5741, pp. 66–76. Springer, Heidelberg (2009)
23. Keßler, C., Raubal, M., Wosniok, C.: Semantic rules for context-aware geographical information retrieval. In: Barnaghi, P., Moessner, K., Presser, M., Meissner, S. (eds.) EuroSSC 2009. LNCS, vol. 5741, pp. 77–92. Springer, Heidelberg (2009)
24. Calero, J.M.A., Ortega, A.M., Perez, G.M., Blaya, J.A.B., Skarmeta, A.F.G.: A nonmonotonic expressiveness extension on the semantic web rule language. J. Web Eng. **11**(2), 93–118 (2012)
25. Mileo, A., Abdelrahman, A., Policarpio, S., Hauswirth, M.: StreamRule: a nonmonotonic stream reasoning system for the semantic web. In: Faber, W., Lembo, D. (eds.) RR 2013. LNCS, vol. 7994, pp. 247–252. Springer, Heidelberg (2013)
26. Le-Phuoc, D., Dao-Tran, M., Xavier Parreira, J., Hauswirth, M.: A native and adaptive approach for unified processing of linked streams and linked data. In: Aroyo, L., Welty, C., Alani, H., Taylor, J., Bernstein, A., Kagal, L., Noy, N., Blomqvist, E. (eds.) ISWC 2011, Part I. LNCS, vol. 7031, pp. 370–388. Springer, Heidelberg (2011)
27. Gebser, M., Grote, T., Kaminski, R., Obermeier, P., Sabuncu, O., Schaub. T.: Answer set programming for stream reasoning. In: CoRR (2013)