Research Chapters in the area of Stream Reasoning

E. Della Valle[1], S. Ceri[1], D. Braga[1], I. Celino[2], D. Frensel[3], F. van Harmelen[4], and G. Unel[3]

[1] Dip. di Elettronica e Informazione, Politecnico di Milano, Via Ponzio, 34/5 - 20133 Milano, Italy
[2] CEFRIEL, via Fucini, 2 - 20133 Milano, Italy
[3] STI-Innsbruck, Technikerstrae 21a, 6020 Innsbruck, Austria
[4] Vrije Universiteit Amsterdam, de Boelelaan 1081a, 1081HV Amsterdam, The Netherlands

Abstract. Data streams occur in a variety of modern applications. Specialized Stream Database Management Systems proved to be an optimal solution for on the fly analysis of data streams, but they cannot perform complex reasoning tasks that requires to combine the streaming data with less time variant knowledge. At the same time, while reasoners are year after year scaling up in the classical, time invariant domain of ontological knowledge, reasoning upon rapidly changing information has been neglected or forgotten. We hereby propose stream reasoning - an unexplored, yet high impact, research area - as the new multi-disciplinary approach which will provide the abstractions, foundations, methods, and tools required to integrate data streams and reasoning systems. In particular the focus of this paper is to sketch the research chapters of Stream Reasoning.

1 Introduction

"Is a traffic jam going to happen in this highway? And is then convenient to reallocate travelers based upon the forecast?" "By looking at the click stream coming from a given IP, can we notice the shifts of interest of the person behind the computer?" "Which contents of the news Web portal are attracting more attention? Which navigation pattern would lead readers to other news related to those contents?" "Are trends in medical records indicative of any new disease spreading in given parts of the world?" "Where are all my friends meeting?" "In the financial context, can we detect any intraday correlation clusters among stock exchange?" Although the information is often available, there's no software system capable of computing the answers - indeed, no system enables users even to issue such queries.

Data streams occur in a variety of modern applications, such as network monitoring, traffic engineering, sensor networks, RFID tags applications, telecom call records, medical records, financial applications, Web logs, click-streams. Specialized Stream Database Management Systems exist. While such systems proved to be an optimal solution for on the fly analysis of data streams, they cannot perform complex reasoning tasks, such as the ones required for computing the answers to the above queries. At the same time, while reasoners are year after year scaling up in the classical, time invariant domain of ontological knowledge, reasoning upon rapidly changing information has been neglected or forgotten. Reasoning systems assume static knowledge, and do not manage "changing worlds" - at most, one can update the ontological knowledge and then repeat the reasoning tasks. We hereby propose stream reasoning - an unexplored, yet high impact, research area - as the new multi-disciplinary approach which will provide the abstractions, foundations, methods, and tools required to integrate data streams and reasoning systems, thus giving answer to the above and innumerable other questions. The idea is simple, yet pervasive.

In order to understand the research chapters that are currently under investigation, we organized the Stream Reasoning 2009 (SR2009) workshop[5] co-located with the European Semantic Web Conference 2009[6]. The main objective of this paper is to provide readers with key to systematically read the paper accepted to SR2009 workshop[1–5] and few others in the field [6–8].

The rest of the paper is organized as follows. We first present, in Section 2, two concrete examples of Stream Reasoning applications. Then, in Section 3, we introduces the problem Stream Reasoning research aim at solving. Section 4 is the central part of this paper and presents a list of research chapters that we believe should be investigated in order to turn Stream Reasoning into a solid reality. With Section 5 and 6, we present a simple, yet effective, approach to measure progress in the area and we draw some conclusions.

2 Concrete Examples of Stream Reasoning Applications

We begin this paper with a list of questions in disparate domains that can be easily answered applying methods and tools resulting from investigation in the area of Stream Reasoning. Hereafter, we provide more details about two concrete examples of Stream Reasoning applications.

2.1 Mobile Applications

Today we live in a world where mobility is a core and always present concept that permeates our lives. Technology comes into place to support and accompany our mobility in several ways, with portable devices, both for business and entertainment. Mobile phones have become so popular and widespread that several applications for mobile phones are being developed in very different areas and with various purposes.

Mobile applications are therefore quite a generic and suitable case for the concept of Stream Reasoning. Being immersed in our everyday life, within our

[5] `http://streamreasoning.org/events/SR2009`
[6] `http://www.eswc2009.org/`

experience, those mobile applications must fulfill real time requirements, especially if they are used to take short-term decisions (like where to go, which means of transportation to choose, which restaurant to select, ...). Using data from sensors, which are likely to come in streams, those mobile applications must find an answer to the problems of reasoning with streams: coping with noise data, dealing with errors, computing the "heavy" reasoning on the server rather that on the mobile devices, etc. Dealing with the "stream of experience" of users, those mobile applications must reason on what part of the streaming information is relevant and what's its "meaning" (e.g. abstracting from quantitative information about latitude and longitude to qualitative information about common places like home, office, gym, etc.). Using mobile phone users as "sensors", those mobile applications could be used also to understand the urban environment and its structure.

2.2 Monitoring of Public Health Risks

Early detection of potentially threatening public health events such as outbreaks and epidemics is a major priority of national and international health related organizations. Examples from the recent past are new infections such as SARS or the H5N1 "bird flu". Dealing with this priority requires the advancement of early detection capabilities, by enabling more timely and thorough acquisition of relevant data and by advancing technologies associated with near real-time reporting and automated outbreak identification. This requires an integrated public health event detection platform that monitors a large variety of heterogeneous distributed data streams for detecting events and situations that might, when interpreted in the appropriate context, signify a potential threat for public health. Such a dynamic platform must identify, integrate and interpret heterogeneous distributed data streams, with information flowing from these data sources automatically analysed and expressed on the basis of rich background knowledge. In the event that the outcome of this process is an increased estimated probability of a threat, notifications to public health bodies will have to be streamlined over various communication channels (e.g. email, mobile phones) and will have to deliver traces of the reasoning process and data that lead to the calculation of the increased threat probability, in order to be evaluated and utilized appropriately. Existing systems such as Google's by now classical "flutrends" do indeed process high volume streams of data, but all semantic processing of this data is done either a priori (integration of streams) or a posteriori (interpretation of results). The challenge is to make the transition from such handcrafted systems to automatic reasoning over data-streams of similar magnitudes.

3 Problem to be Solved

The areas that can be positively impacted by Stream Reasoning are numerous. Finance, energy supply management, attention mining for Web 2.0 application, traffic management, real-time social-networking, healthcare are just a few of those areas. Some years ago, proposing to develop a system to answer questions like the one above would have looked like a Sci-Fi idea due to the lack of data. Nowadays, a large amount of the required information are already available in digital format and can be access at almost no cost: maps with the commercial activities and meeting places, events scheduled in the city and their locations, average speed in highways, positions and speed of public transportation vehicles, parking availabilities in specific parking areas, geo-positioned twitter posts, user generated media of any kind, web logs, click streams, epidemiological data, as so on and so forth. The problem is that current technologies are not up to the challenges to reason upon all this rapidly changing information. To do so, a system requires coping with:

1. *heterogeneity* both in data stream sources and in static information sources at syntactic, structural and semantic level;
2. *time dependencies*, since the very nature of stream, data is valuable only when it is actually presented; if it is not captured and immediately summarized, then reconstructing the value is impossible - of course, all the information is also subject to change through classical update mechanisms;
3. *window dependencies*, since data are observed trough a window, which can span in time or in number of elements it can contain, information about individuals in a given time window can be either incomplete (e.g., some sensors did not provide data) or over constrained (e.g., different sensors observing the same event);
4. *noisy and uncertain data*, i.e. data coming from a sensor network in a given moment may be faulty due to faults in some sensors or in part of the network;
5. *scale*, i.e., both the presence of huge data throughputs and the need to link streaming data with static knowledge, where perhaps only very limited amount data and knowledge are sufficient for a given reasoning tasks and the data should therefore be identified, sampled, abstracted and approximated;
6. *real-time constraints*, i.e., an answer should be provided before it becomes useless, which leads to the need for incremental query answering and reasoning;
7. *continuous processing*, applications are either interested into fresh data - thus, if they lose the data stream, they totally lose their relevance - or into summary data - but again, once that summarization is needed, it is much more rationale doing it once and for all by optimizing the continuous data processing than doing independent summarization upon masses of persistent data. Thus, continuous query processing performs the optimization by combining summarization requirements all at once, and then lets the irrelevant data (perhaps 99.99%) to get lost; and
8. *distribution of computational units*, which also means modularizing the reasoning, minimizing the transmission data among the units and being able to control the reasoning process.

4 Research Chapters

By systematically analyzing the problems presented in Section 3, we were able to divide the Stream Reasoning research in 5 chapters.

4.1 Theory for Stream Reasoning

Stream Reasoning research definitely need new theoretical investigations that go beyond Data Stream Management Systems [9], Event based system [10] and Complex Event Processing [11].

Examples of important theoretical problems that need investigations are:

- Dealing with incomplete or over constrained information about individuals as proposed in [5],
- Notion of symbol grounding as referred in [2], and
- Notion of soundness and completeness for stream reasoning.

4.2 Logic language for stream reasoning

Investigations about which logic language is appropriate for stream reasoning is an important theoretical aspect; therefore we dedicate to it a separate research chapter.

The paper submitted to SR2009 adopt a variety of different logics. A Constructive Description Logic [12] is at the core of [5]. A Commonsense Spatial Hybrid Logics [13] is proposed in [1]. Metric Temporal Logic [14] is the logical language of the DyKnow middleware [2]. Indeed several other logics, which appear to be valid starting points, exists; e.g., Temporal Action Logic [15], Step Logic [16] and Active Logic [17].

4.3 Stream Data Management for the Semantic Web

A first step toward Stream Reasoning is certainly trying to combine the power of existing Data Stream Management Systems and existing reasoning techniques. The key idea is to keep streaming data in relational format as long as possible and bring them at the reasoning level as aggregated events [18]. Even to do so, existing data models and query languages for Data Stream Management Systems and reasoners are not sufficient; they must be combined. A simple notion of RDF stream and a basic extension to SPARQL (named Streaming SPARQL) is proposed in [7]. A more complete proposal (named C-SPARQL), which includes aggregate and timestamp functions, is presented in [8]. The Knowledge Processing Language presented in [2] also provides a way to represent and query streams.

However, more investigation is need for Query Execution and Optimization. Interesting research topics appear to be:

- cost metrics to measure query plan cost,

– continuous query plan adaptation to the bursty nature of data streams,
– parallel processing of multiple queries to exploit inter-query optimization opportunities, and
– distributed query processing.

4.4 Stream Reasoning for the Semantic Web

Combining Stream Data Management and reasoning at data model and query language level is only a first step toward Stream Reasoning, a deeper merge can be investigated. From different view points, part of this research has been conducted in Artificial Intelligence under the name of belief revision [19], however a well developed notion of Stream Reasoning has not been proposed yet.

The central research question is: can the idea of continuous semantics introduced in Data Stream Management System be extended to reasoners? For instance, can materializations be incrementally maintained? But even more basically, do the current materialization hold? How long will it? Can an updated materialization be computed before it will be outdated? Last but not least, can Stream Reasoning benefit from distribution and parallelization?

We find very interesting the attempts to answer this questions the incremental evaluation of complex temporal formula described in [2] and incremental answering of reachability queries on streaming graphs described in [3].

4.5 Stream Reasoning Engineering

Engineering of Stream Reasoning is clearly in its infancy. Several implemented systems exists (e.g., [2, 1, 6], but a systematic approach was only attempted in DyKnow [2], which introduces notion of primitive streams, stream generator, stream consumer and stream processor, and in [18], which applies to data streams the concept of identification, selection, abstraction and reasoning proposed in the LarKC approach [20]. Investigating a Conceptual Architecture for Stream Reasoning is clearly needed.

Moreover, all the research problems listed in the chapter above need some degree of engineering. Hereafter we list the key engineering activities needed to develop a solid implementation of Stream Reasoning:

– Integration of data streams with reasoning systems,
– Optimization methods for Stream Reasoning,
– Scalability issues in stream reasoning,
– Real time reasoning,
– Approximate stream reasoning,
– Distribution issues in stream reasoning, and
– Evaluation of stream reasoners.

4.6 Application of Stream Reasoning

Application of Stream Reasoning deserve a research chapters on their own, because the idea of Stream Reasoning does not arise as a theoretical research topic, even if requires major theoretical researches, but as a potential solution to real problems. Traffic Monitoring and traffic pattern detection appears to be a very natural area, since it was independently studied in [1–3, 18]. Other area of interest are financial transaction continuous auditing [5], wind power plant monitoring [7], situation-aware mobile services [4] and patient monitoring systems [6]. We believe that other areas of investigation, characterized by an high impact, can be: Web blogs monitoring (see Section sec:cases-ph), click streams real-time analysis and mobile social networking.

5 Measuring Progress

Although the problem may appear intractable at first glance, a roadmap for Stream Reasoning can be sketch as follows. Once one accepts that no Stream Reasoning is possible in the space of the *onetime semantics* of standard reasoning and thus it is only possible when thinking in terms of *continuous semantics*, then the system must have the notion of *observation period*, defined as the period when the system is subject to querying. In current reasoners, all forms of knowledge are invariable and data can be updated, but they are not allowed to change too frequently. The notion of observation period together with a classification of what kind of knowledge and data is allow to change allow ordering progressively more complex form of stream reasoning. The table below presents our intuition and it can be used to measure progress.

	Level of Complexity				
Kind of Knowledge	Low	Medium	High	Very High	Extream
Terminological Knowledge	Invariable	Invariable	Invariable	Invariable	Allowed
Nomological Knowledge	Invariable	Invariable	Invariable	Allowed	Allowed
Factual Knowledge	Invariable	Invariable	Allowed	Allowed	Allowed
Event-driven Changing Data	Invariable	Allowed	Allowed	Allowed	Allowed
Streaming Data	Allowed	Allowed	Allowed	Allowed	Allowed

Table 1. The kind of knowledge which is allowed to changed within the observation period provide a simple, yet effective, wa to identify four different level of complexity in Stream Reasoning.

All the researches, which we have been discussing in this paper and which prototyped a working system [1, 2, 6–8], ground the stream reasoning core model upon known database and reasoning methods. It's clear that the adoption of off-the-shelf stream database and reasoning tools provide both a solid framework and a fast way for prototyping.

6 Conlusion

While the works discussed in this paper serve to ground stream reasoning and to give an intuition that the task is not impossible, a huge amount of innovation is required in order to cover the queries that we have initially set as our ambitious target and thus covering the gap between the current state-of-the-art to bring stream reasoning into life.

Starting from lesson learned in the database community (e.g., the ability to efficiently abstract and aggregate information out of multiple, high-throughput streams) a new foundational theory of stream reasoning is needed, capable to associate reasoning tasks to time windows describing data validity and to therefore to produce time-varying inferences. From these foundations, new paradigms for knowledge representation and query languages design must be derived, and the consequent computational frameworks for stream reasoning oriented software architectures and their instrumentation must be deployed.

Acknowledgments

The work described in this paper is has been partially supported by the European project LarKC (FP7-215535).

References

1. Palmonari, M., Bogni, D.: Commonsense spatial reasoning about heterogeneous events in urban computing. In: Stream Reasoning. (2009)
2. Heintz, F., Kvarnstrom, J., Doherty, P.: Stream reasoning in dyknow: A knowledge processing middleware system. In: Stream Reasoning. (2009)
3. Unel, G., Fischer, F., Bishop, B.: Answering reachability queries on streaming graphs. In: Stream Reasoning. (2009)
4. Luther, M., Bhm, S.: Situation-aware mobility: An application for stream reasoning. In: Stream Reasoning. (2009)
5. Mendler, M., Scheele, S.: Towards a type system for semantic streams. In: Stream Reasoning. (2009)
6. Ordonez, P., Kodeswaran, P.B., Korolev, V., Li, W., Walavalkar, O., Elgamil, B., Joshi, A., Finin, T., Yesha, Y., George, I.: A ubiquitous context-aware environment for surgical training. In: MobiQuitous. (2007) 1–6
7. Bolles, A., Grawunder, M., Jacobi, J.: Streaming sparql - extending sparql to process data streams. In: ESWC. (2008) 448–462
8. Barbieri, D.F., Braga, D., Ceri, S., Valle, E.D., Grossniklaus, M.: C-sparql: Sparql for continuous querying. In: WWW. (2009) 1061–1062
9. Garofalakis, M., Gehrke, J., Rastogi, R.: Data Stream Management: Processing High-Speed Data Streams (Data-Centric Systems and Applications). Springer-Verlag New York, Inc., Secaucus, NJ, USA (2007)
10. Mhl, G., Fiege, L., Pietzuch, P.: Distributed Event-Based Systems. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2006)

11. Luckham, D.: The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems. Springer-Verlag New York, Inc., Secaucus, NJ, USA (2008)
12. Mendler, M., Scheele, S.: Towards constructive dl for abstraction and refinement. In: Description Logics. (2008)
13. Bandini, S., Mosca, A., Palmonari, M.: Common-sense spatial reasoning for information correlation in pervasive computing. Applied Artificial Intelligence **21**(4&5) (2007) 405–425
14. Ouaknine, J., Worrell, J.: Some recent results in metric temporal logic. In: FORMATS. (2008) 1–13
15. Doherty, P., Gustafsson, J., Karlsson, L., Kvarnström, J.: Tal: Temporal action logics language specification and tutorial. Electron. Trans. Artif. Intell. **2** (1998) 273–306
16. Elgot-Drapkin, J.J.: Step-logic: reasoning situated in time. PhD thesis, College Park, MD, USA (1988) Director-Perlis,, Donald.
17. Elgot-drapkin, J., Kraus, S., Miller, M., Nirkhe, M., Perlis, D.: Active logics: A unified formal approach to episodic reasoning (1999)
18. Valle, E.D., Ceri, S., Barbieri, D.F., Braga, D., Campi, A.: A first step towards stream reasoning. In: FIS. (2008) 72–81
19. Darwiche, A., Pearl, J.: On the logic of iterated belief revision. Artif. Intell. **89**(1-2) (1997) 1–29
20. Fensel, D., van Harmelen, F., Andersson, B., Brennan, P., Cunningham, H., Della Valle, E., Fischer, F., Huang, Z., Kiryakov, A., il Lee, T.K., School, L., Tresp, V., Wesner, S., Witbrock, M., Zhong, N.: Towards larkc: a platform for web-scale reasoning, IEEE International Conference on Semantic Computing (ICSC 2008) (2008)