Future Generation Computer Systems ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Automated discovery and integration of semantic urban data streams: The ACEIS middleware

Feng Gao [a,b,c,*], Muhammad Intizar Ali [c], Edward Curry [c], Alessandra Mileo [c,d]

[a] Department of Computer Science, Wuhan University of Science and Technology, China
[b] Hubei Province Key Laboratory of Intelligent Information Processing and Real-time Industrial System, China
[c] Insight Centre for Data Analytics, National University of Ireland, Galway, Ireland
[d] Insight Centre for Data Analytics, Dublin City University, Ireland

HIGHLIGHTS

- We present the architecture of the Automated Complex Event Implementation System.
- We introduce a Complex Event Service (CES) Ontology, and demonstrate its usage.
- We define the formal semantics of the event in CES and align it with RSP semantics.
- We implement a query transformation system to create RSP queries from CES annotations.
- We show the usage of ACEIS in Smart City and optimize its capacity for concurrent users.

ARTICLE INFO

Article history:
Received 15 April 2016
Received in revised form
8 December 2016
Accepted 1 March 2017
Available online xxxx

Keywords:
Semantic Web
Complex events
Service computing
RDF Stream Processing

ABSTRACT

With the growing popularity of Internet of Things (IoT) technologies and sensors deployment, more and more cities are leaning towards smart cities solutions that can leverage this rich source of streaming data to gather knowledge that can be used to solve domain-specific problems. A key challenge that needs to be faced in this respect is the ability to automatically discover and integrate heterogeneous sensor data streams on the fly for applications to use them. To provide a domain-independent platform and take full benefits from semantic technologies, in this paper we present an Automated Complex Event Implementation System (ACEIS), which serves as a middleware between sensor data streams and smart city applications. ACEIS not only automatically discovers and composes IoT streams in urban infrastructures for users' requirements expressed as complex event requests, but also automatically generates stream queries in order to detect the requested complex events, bridging the gap between high-level application users and low-level information sources. We also demonstrate the use of ACEIS in a smart travel planner scenario using real-world sensor devices and datasets.

1. Introduction

An increasing number of cities have started to embrace the idea of smart cities and are in the process of building smart city infrastructure for their citizens [1]. Such infrastructures, including sensors, open data platforms and smart city applications, can improve the day to day life for the citizens. A typical example of smart city applications is the provision of real-time tracking and timetable information for the public transport within the city.[1] The city of Aarhus provides an open data platform called ODAA,[2] which contains city related information generated by various sensors deployed within the city, e.g., traffic congestion level, air quality and trash-bin level etc. ODAA also encourages usage of their open data platform for building smart city applications. In the foreseeable future, more and more urban data will be made available. The enormous amount of data produced by sensors in our day to day life needs to be harnessed to help smart city applications taking smart decisions on-the-fly.

However, despite the increasing amount of infrastructures and datasets available, the uptake of smart city applications is

* Corresponding author at: Room 301, Unit 3, Building 9; Yang Guang Xin Yuan; JiangHan District, Wuhan City, China.
E-mail addresses: feng.gao86@wust.edu.cn (F. Gao), ali.intizar@insight-centre.org (M.I. Ali), edward.curry@insight-centre.org (E. Curry), alessandra.mileo@insight-centre.org (A. Mileo).

http://dx.doi.org/10.1016/j.future.2017.03.002
0167-739X/© 2017 Published by Elsevier B.V.

[1] Live bus arrivals in London: http://countdown.tfl.gov.uk/#/.
[2] Open Data Aarhus: http://odaa.dk.

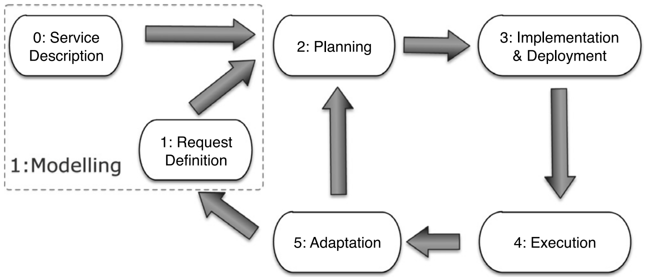

Fig. 1. Event service life-cycle.

hindered by various issues, such as the difficulty of discovering the capabilities of the available infrastructure and once discovered, integrating heterogeneous data sources and extracting up-to-date, reliable information in real-time. Complex Event Processing (CEP) [2,3] has matured from the last few decades that aggregates low-level data and provide abstracted high-level information. Recently, semantic event processing and RDF Stream Processing (RSP) [4,5] have been studied to bring semantics into CEP and deal with the data heterogeneity. Like most CEP solutions, existing RSP engines assume the streams used in the queries are identified and do not address the problem of discovering proper stream sources on-demand. Moreover, various RSP platforms have been created but a unified RSP syntax and semantics have yet to be established [6], and hence, collaborations between different RSP platforms are difficult. We plan to address this issue by providing RSP capabilities as semantically described services and aligning the formal semantics of different RSP engines.

We choose the service-oriented paradigm for enabling a collaborative, on-demand and cross-platform RSP, mainly because this way we can decouple RSP providers and consumers. Semantic Web Service (SWS) have been discussed extensively in service computing. SWS transcends conventional Web Services by applying Semantic Web techniques to realise automatic service discovery and composition [7]. However, existing SWS approaches do not cater complex event services While existing semantic service discovery and composition approaches (e.g., WSMO,[3] OWL-S[4]) show great potentials in service discovery and composition compared to syntactical service discovery [7], they are based on *Input, Output, Precondition and Effect*, e.g., in [8]. However, the functionalities of event services are determined by the semantics of the events they deliver, which is captured by the event patterns defined within an event algebra [9]. A pattern-based composition is needed for complex event services, which is not available in state-of-the-art service composition mechanisms. Apparently we are not the first that try to enable service-oriented event processing. In [10] an Enterprise Service Bus (ESB) based architecture was proposed. We essentially seek to address a similar problem as in [10], but in the context of RSP rather than conventional CEP.

In this paper, we present the Automatic Complex Event Implementation System (ACEIS), which is an automated discovery and integration system for urban data streams. We design a semantic information model to represent complex event services (as an extension of OWL-S ontology) and utilise this information model for the discovery and integration of sensor data streams. ACEIS assumes that all available sensor data streams are annotated using the Semantic Sensor Network (SSN) ontology[5] and stored in a repository. Various Quality of Service (QoS) and Quality of Information (QoI) metrics are also annotated for each sensor data stream.

ACEIS receives an event service request described using our complex event service information model and automatically discovers and composes the most suitable data streams for the particular event request. ACEIS then transforms the event service composition into a stream query to be deployed and executed on a stream engine to evaluate the complex event pattern specified in the event service request. In summary, ACEIS is a middleware for managing the life cycle of event services, which includes the modelling, planning, implementation, execution and adaptation. Fig. 1 illustrates the life cycle of event services (by the analogy to Web Service life cycle). Our previous work have discussed the modelling [11], planning [12] and adaptation [13] aspects. In this paper, we present the big picture of ACEIS to show how different parts come together, with a focus on how the implementation is carried out for event services, and how the execution can be optimised. The contributions of this paper can be summarised as below:

* We present our Automated Complex Event Implementation System serving as a middleware between Smart City applications and sensor data streams and we provide an overview of its components and their interactions (Section 4).
* We describe the formal semantics of the event patterns in CES and compare it with the query semantics of semantic event processing systems to ensure a correct query transformation and evaluation (Section 6).
* We implement an automatic query transformation system to formulate continuous queries over semantic sensor data streams based on the alignment of event and query semantics (Section 7).
* We demonstrate how ACEIS is used in a Smart City Application scenario and provide evaluation and optimisation for the capacity of ACEIS, with regard to handling concurrent user queries (Sections 8, 9).

Structure of the paper: In Section 2 we introduce the background of our work (including RDF Stream Processing and Semantic Web Service) and then compare our work with the state-of-the-art. In Section 3, we present some Smart City scenarios, together with various types of sensor data streams that can be used in these scenarios as well as the challenges faced by smart city applications. We present the overall architecture of our system (ACEIS) in Section 4. A brief description of the sensor data streams discovery and integration is provided in Section 5. Section 6 lays down the formal semantics of the complex events modelled in ACEIS. Section 7 discusses our automated query transformation algorithm based on the event semantics and stream query semantics. Section 9 discusses the optimisation techniques for handling concurrent queries in ACEIS, before concluding in Section 10.

Before we move on to the next section, we provide the definitions of the terms used in this paper in Table 1.

2. Related work

In this paper, we focus on providing on-demand, cross-platform RSP using Service Oriented Architecture. In this section, we first introduce RSP and SWS as the context of our work. Publish–Subscribe Systems are also relevant for this paper, since they also discuss how different event processing results can be

3 Web Service Modeling Ontology: http://www.wsmo.org/.
4 OWL-S ontology: http://www.w3.org/Submission/OWL-S/.
5 SSN ontology: http://www.w3.org/2005/Incubator/ssn/ssnx/ssn.

Please cite this article in press as: F. Gao, et al., Automated discovery and integration of semantic urban data streams: The ACEIS middleware, Future Generation Computer Systems (2017), http://dx.doi.org/10.1016/j.future.2017.03.002