Computers in Industry 94 (2018) 82–97

Contents lists available at ScienceDirect

Computers in Industry

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

Oliot EPCIS: Engineering a web information system complying with EPC Information Services standard towards the Internet of Things

Jaewook Byun*, Sungpil Woo, Yalew Tolcha, Daeyoung Kim

School of Computing, Korea Institute of Science and Technology (KAIST), 291 Daehak-ro, Yuseong-gu, Daejeon 34141, South Korea

ARTICLE INFO

Article history:
Received 27 July 2017
Received in revised form 21 September 2017
Accepted 24 October 2017
Available online xxx

Keywords:
Oliot EPCIS
EPCIS
GS1
Supply chain management
The Internet of Things
IoT

ABSTRACT

A global standard, Electronic Product Code Information Services (EPCIS), standardizes ways to capture and share crucial moments in a lifecycle of physical objects. EPCIS has the possibility of becoming a viable standard for a web information system of the Internet of Things (IoT) because EPCIS is a de facto standard for Radio Frequency IDentification (RFID) technology and be flexible enough to manage various sensor data in IoT environments. However, an EPCIS system should become efficient and scalable enough to deal with billions of physical objects. In the paper, we share our three-year experience developing an open source, Oliot EPCIS. The experience in several research projects leads to the continuously refined architecture as well as our extra features for efficiency and scalability. According to the experiments, Oliot EPCIS is more efficient and scalable than existing open source solutions. Also, our discussion of the results shows a user the appropriate way to use Oliot EPCIS in one's own application domain.

1. Introduction

Global Standard 1 (GS1)[1] is an international organization that develops global standards. Most of the GS1 standards focus on how to assign an identifier to a physical object and give visibility to its lifecycle. For example, GS1 ratifies a code system that assigns a globally unique identifier to a Radio Frequency IDentification (RFID) tag attached to a physical object, Electronic Product Code (EPC) [1]. Based on this unique identification system, GS1 ratifies an information service standard for capturing and sharing visibility event data as well as master data, which provide a globally consistent view of physical objects both within and across enterprises, Electronic Product Code Information Service (EPCIS) [2].

We expect that EPCIS can be a viable standard for a web information system of the Internet of Things (IoT). On the one hand, EPCIS is a de facto standard for RFID technology. Identification of physical objects is known as one challenge facing IoT [3,4], and the RFID technology is one of the driving forces enabling the identification [5,6].

On the other hand, an extension mechanism of EPCIS is flexible enough for its scope to broaden to the Internet of Things. EPCIS allows developers to design additional event types not defined in the standard [7–9]; also, EPCIS enables users to insert additional data into an existing event type such as temperature [10], health signal [11], bridge condition [12], geographic longitude/latitude [13], and on-board diagnostics (OBD) values [14].

However, an EPCIS system[2] should meet prerequisites for IoT environments. For example, the system should be efficient and scalable enough to afford the plethora of requests from billions of devices [15].

In the paper, we share our three-year experience engineering an open source EPCIS system, Oliot EPCIS.[3] In several research projects, we have applied the system to healthcare, civil engineering, automobile, agriculture, etc. The requirements from each application domain have continuously refined the architecture and make us provide extra features.

We expect that the architecture rids EPCIS developers of the inefficient system development (e.g., trigger subscription) and facilitates the development of efficient and scalable systems (e.g., parallelism). Also, it is possible that the architecture promotes more universal usage of EPCIS by alleviating the implementation complexity [16,17]. Meanwhile, the extra features provide the ways to interact with a large number of resource-constrained devices

* Corresponding author.
E-mail addresses: bjw0829@kaist.ac.kr (J. Byun), woosungpil@kaist.ac.kr (S. Woo), yalewkidane@kaist.ac.kr (Y. Tolcha), kimd@kaist.ac.kr (D. Kim).

[1] GS1, Web Page, https://www.gs1.org/.

[2] In the paper, EPCIS indicates the standard and an EPCIS system indicates a system that implements the standard-compliant interfaces. The version of EPCIS discussed in the paper is EPCIS v1.2 ratified in September 2016.

[3] Oliot EPCIS, GitHub Page, https://github.com/JaewookByun/epcis.

https://doi.org/10.1016/j.compind.2017.10.004
0166-3615/© 2017 Elsevier B.V. All rights reserved.

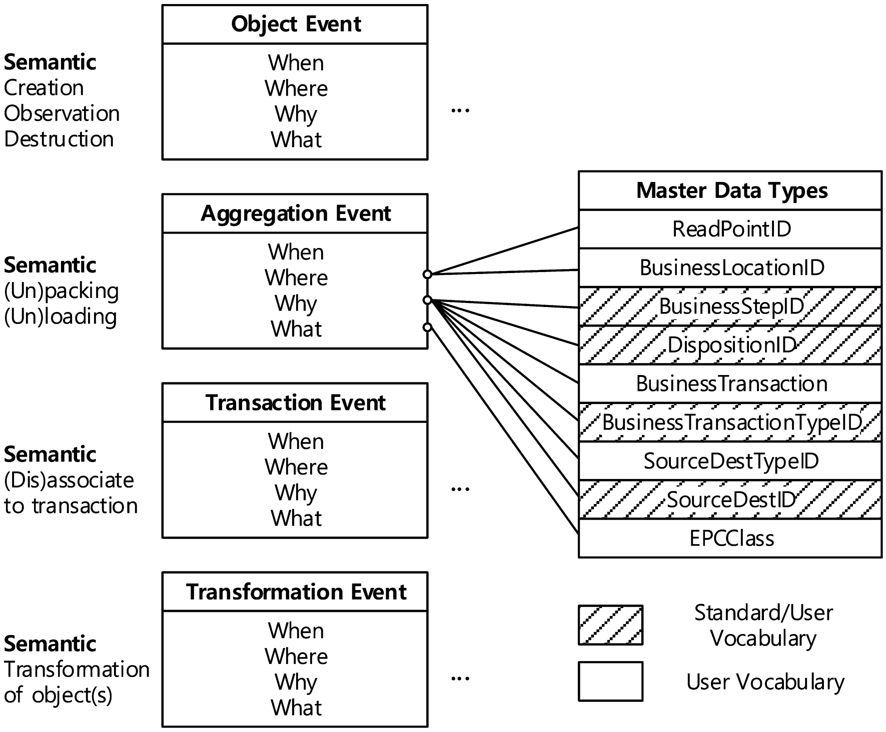

Fig. 1. EPCIS standardizes four event types where each of them is suitable for representing its respective crucial moments in a lifecycle. Most of the contents in events are encoded as Uniform Resource Name (URN) where detailed information of each URN is managed as master data.

(e.g., lightweight message format, lightweight service interface, and projection of query results). We believe that these features are general enough to be included in the standard.

The remainder of the paper is organized as follows. In Section 2, we introduce EPCIS. Then, Section 3 discusses the design decisions towards IoT environments. Next, we present a system architecture and its implementation in Section 4. Section 5 evaluates Oliot EPCIS with existing solutions and discusses the effect of our design decisions and extensions with extensive experiments. Finally, we review the related works and conclude the paper.

2. Overview of EPC Information Services (EPCIS) standard

2.1. Data model

There are two kinds of data treated in EPCIS: event data and master data. Event data is temporal data, which stands for a situation of physical objects at a specific time while master data provides the holistic information of physical objects not relevant to a specific time such as name and owner.

Fig. 1 visualizes a data model of EPCIS. There are four types of standardized events where each type is suitable for expressing its respective crucial moments in a lifecycle. For example, *Object Event* is a commonly used event type that represents creation/observation/destruction of physical objects. In supply chains, a tag-attaching physical object becomes visible on the Internet through RFID or barcode reader. In its typical lifecycle, an EPC is first assigned to a newly created physical object; then, the object is observed in various places; finally, the EPC is disassociated from the object when the object is destroyed.

Most of the contents in events (i.e., *where*, *why* and *what*) are encoded as a Uniform Resource Name (URN) where detailed information of each URN is managed as master data. Some of the master data, called *standard vocabulary*, are defined in Core Business Vocabulary (CBV) [18]. CBV provides a definition and examples of a URN (e.g., *urn:epcglobal:cbv:bizstep:loading*) for the global consensus of the vocabulary. Meanwhile, users can define their own vocabularies, called *user vocabulary*, to store detailed information of URNs. For example, people can define a business location vocabulary, *urn:epc:id:sgln:0614141.07346.1234*, and store its address, name, geographic coordinates, etc. Then, people can access to the detailed information in EPCIS repositories.

A set of event data and master data is enveloped in a document, EPCIS document. The format of EPCIS document is forced to Extensible Markup Language (XML), and EPCIS provides several XML Schema Definition (XSD) files[4] to define the structure of EPCIS Document.

2.2. Service interface

EPCIS standardizes the ways to capture and share EPCIS documents. Among eight service interfaces, we summarize three core interfaces as shown in Table 1. The *capture* interface receives an EPCIS document and stores event/master data inside the document into an EPCIS repository.

EPCIS defines two sets of query parameters to retrieve the stored event data (i.e., *SimpleEventQuery*) and the stored master data (i.e., *SimpleMasterDataQuery*). Basically, if no query parameter provided, EPCIS returns all of the stored event/master data; then, query parameters enable to filter/sort/limit the returnable data. For example, *GE_eventTime*, a filter parameter, returns the event data occurred after the specific time; *orderBy*, a sorting parameter, sorting the results by a time or a user extended value; *eventCountLimit* limits the size of the events to be returned.

Users can obtain the data from the repository immediately through the *poll* method. Meanwhile, event data can be delivered to a specific destination periodically (i.e., *scheduled subscription*) or

[4] EPCIS 1.2 Artifacts, Web Page, https://www.gs1.org/epcis/epcis-artifacts/1-2.