Journal of Network and Computer Applications 64 (2016) 137–153

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

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

When things matter: A survey on data-centric internet of things

Yongrui Qin [a,*], Quan Z. Sheng [a], Nickolas J.G. Falkner [a], Schahram Dustdar [b], Hua Wang [c], Athanasios V. Vasilakos [d]

[a] School of Computer Science, The University of Adelaide, Adelaide SA 5005, Australia
[b] Institute of Information Systems, Vienna University of Technology, Austria
[c] Center for Applied Informatics, Victoria University, Melbourne, Victoria 3122, Australia
[d] Department of Electrical and Computer Engineering, National Technical University of Athens, Greece

ARTICLE INFO

Article history:
Received 18 August 2014
Received in revised form
22 June 2015
Accepted 28 December 2015
Available online 11 February 2016

Keywords:
Internet of Things
Data management
RFID systems
Wireless sensor networks

ABSTRACT

With the recent advances in radio-frequency identification (RFID), low-cost wireless sensor devices, and Web technologies, the Internet of Things (IoT) approach has gained momentum in connecting everyday objects to the Internet and facilitating machine-to-human and machine-to-machine communication with the physical world. IoT offers the capability to connect and integrate both digital and physical entities, enabling a whole new class of applications and services, but several significant challenges need to be addressed before these applications and services can be fully realized. A fundamental challenge centers around managing IoT data, typically produced in dynamic and volatile environments, which is not only extremely large in scale and volume, but also noisy and continuous. This paper reviews the main techniques and state-of-the-art research efforts in IoT from data-centric perspectives, including data stream processing, data storage models, complex event processing, and searching in IoT. Open research issues for IoT data management are also discussed.

1. Introduction

The Internet is a global system of networks that interconnect computers using the standard Internet protocol suite. It has significant impact on the world as it can serve billions of users worldwide. Millions of private, public, academic, business, and government networks, of local to global scope, all contribute to the formation of the Internet. The traditional Internet has a focus on computers and can be called the Internet of Computers. In contrast, evolving from the Internet of Computers, the Internet of Things (IoT) emphasizes things rather than computers (Ashton, 2009). It aims to connect everyday objects, such as coats, shoes, watches, ovens, washing machines, bikes, cars, even humans, plants, animals, and changing environments, to the Internet to enable communication/interactions between these objects. The ultimate goal of IoT is to enable computers to see, hear and sense the real world. It is predicted by Ericsson that the number of Internet-connected things will reach 50 billion by 2020. Electronic devices and systems exist around us providing different services to the people in different situations: at home, at work, in their office, or driving a car on the street. IoT also enables the close relationship between human and opportunistic connection of smart things (Guo et al., 2013).

There are several definitions or visions of IoT from different perspectives. From the viewpoint of services provided by things, IoT means "a world where things can automatically communicate to computers and each other providing services to the benefit of the human kind" (CASAGRAS, 2000). From the viewpoint of connectivity, IoT means "from anytime, anyplace connectivity for anyone, we will now have connectivity for anything" (ITU, 2005). From the viewpoint of communication, IoT refers to "a world-wide network of interconnected objects uniquely addressable, based on standard communication protocols" (INFSO, 2008). Finally, from the viewpoint of networking, IoT is the Internet evolved "from a network of interconnected computers to a network of interconnected objects" (European Commission, 2009).

We focus on our study of the Internet of Things from a *data perspective*. As shown in Fig. 1, data is processed differently in the Internet of Things and traditional Internet environments (i.e., Internet of Computers). In the Internet of Computers, both main data producers and consumers are human beings. However, in the Internet of Things, the main actors become *things*, which means things are the majority of data producers and consumers. Therefore, we give our definition of the Internet of Things as follows:

"*In the context of the Internet, addressable and interconnected things, instead of humans, act as the main data producers, as well as the main data consumers. Computers will be able to learn and gain information and knowledge to solve real world problems directly with the data fed from things. As an ultimate goal, computers enabled by the Internet of Things technologies will be able to sense and react to the real world for humans.*"

* Corresponding author.
E-mail address: yongrui.qin@gmail.com (Qin).

http://dx.doi.org/10.1016/j.jnca.2015.12.016
1084-8045/

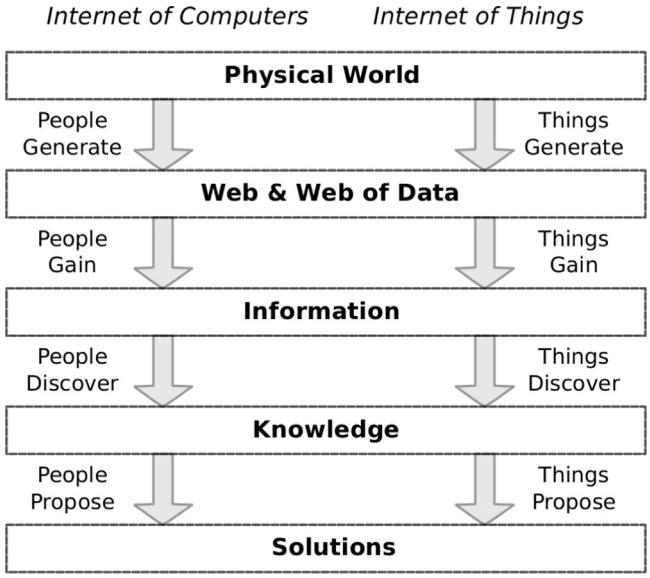

Fig. 1. Internet of Computers vs. Internet of Things.

As of 2012, 2.5 quintillion (2.5×10^{18}) bytes of data are created daily.[1] In IoT, connecting all of the things that people care about in the world becomes possible. All these things would be able to produce much more data than nowadays. The volumes of data are vast, the generation speed of data is fast and the data/information space is global (James et al., 2009). Indeed, IoT is one of the major driving forces for *big data analytics*. Given the scale of IoT, topics such as storage, distributed processing, real-time data stream analytics, and event processing are all critical, and we may need to revisit these areas to improve upon existing technologies for applications of this scale.

In this paper, we systematically investigate the key technologies related to the development of IoT and its applications, particularly from a data-centric perspective. The aim of this work is to provide a better understanding of the current research activities and issues. Fig. 2 shows the roadmap of this paper. As can be seen from the figure, we review and compare technologies including data streams, data storage models, searching, and event processing technologies, which play a vital role in enabling the vision of IoT. We also describe some relevant applications from several representative areas. Although some reviews about IoT have been conducted recently (e.g., Atzori et al., 2010; Zeng et al., 2011; An et al., 2013; Perera et al., 2013; Li et al., 2016; Yan et al., 2014), they focus on high level general issues and are mostly fragmented. In addition, these articles do not specifically cover techniques on data processing and management, which is fundamentally critical to fully embrace IoT. To the best of our knowledge, this is the first article that studies and discusses state-of-the-art techniques of IoT from the data-centric perspective.

The remainder of the article is organized as follows. Section 2 identifies an IoT data taxonomy. Section 3 reviews the data streaming techniques and Section 4 focuses on the data models and storage technologies for IoT. Search and event processing technologies are discussed in Sections 5 and 6, respectively. In Section 7, some typical ongoing and/or potential IoT applications where data techniques for IoT can bring significant changes are described. Finally, Section 8 highlights some research open issues on IoT from the data perspective and Section 9 offers some concluding remarks.

2. IoT data taxonomy

In this section, we identify the intrinsic characteristics of IoT data and classify them into three categories, including *Data Generation, Data Quality*, and *Data Interoperability*. We also identify specific characteristics of each category, and the overall IoT data taxonomy is shown in Fig. 3.

2.1. Data generation

- *Velocity*: In IoT, data can be generated at different rates. For example, for GPS-enabled moving vehicles in road networks, the GPS signal sampling frequency could be every few seconds, every few minutes, or even every half an hour. But some sensors can scan at a rate up to 1,000,000 sensing elements per second.[2] On one hand, it is challenging to handle very high sampling rates, which require efficient processing of the fast generated data. On the other hand, it is also challenging to deal with low sampling rates, due to the fact that some important information may be lost for data processing and decision making.
- *Scalability*: Since things are able to continuously generate data together with the foreseeable excessively large number of things, the IoT data is expected to be at an extremely large scale. It is easy to image that, in IoT data processing systems, scalability will be a long standing issue, aligning with the current Big Data trend.
- *Dynamics*: There are many dynamic elements within IoT data. Firstly, many things are mobile, which will lead to different locations at different times. Since they will move to different environments, the sensing results of things will also be changing to reflect the real world. Secondly, many things are fragile. This means the generated data will change overtime due to the failure of things. Thirdly, the connections between things could be intermittent. This also creates dynamics in any IoT data processing system.
- *Heterogeneity*: There will be many kinds of things potentially connecting to the Internet in the future, ranging from cars, robots, fridges, mobile phones, to shoes, plants, watches, and so on. These kinds of things will generate data in different formats using different vocabularies. In addition, there will be assorted IoT data processing systems, which will also provide data in customized formats to tailor different data needs.

2.2. Data quality

- *Uncertainty*: In IoT, uncertainty may come from different sources. In RFID data, the uncertainty can refer to missing readings, readings of non-existing IDs, etc. In wireless sensor networks, uncertainty can refer to sensing precision (the degree of reproducibility of a measurement), or accuracy (the maximum difference that will exist between the actual value and the indicated value), etc.
- *Redundancy*: Redundancy can also be easily observable in IoT. For example, in RFID data, the same tags can be read multiple times at the same location (because multiple RFID readers exist at the spot or tags are read multiple times at in the same spot) or at different locations. In wireless sensor networks, a group of sensors of the same type may also be deployed in a nearby area, which can produce similar sensing results of that area. For the same sensor, due to the possible high sampling rates, redundant sensing data can be produced.
- *Ambiguity*: Dealing with a large amount of ambiguity in IoT data is inevitable. The data produced by assorted things can be interpreted in different ways due to different data needs from different things or other data consumers. Such data can also be useful and important to

[1] http://www-01.ibm.com/software/data/bigdata/

[2] https://www.tekscan.com/support/faqs/what-are-sensors-sampling-rates

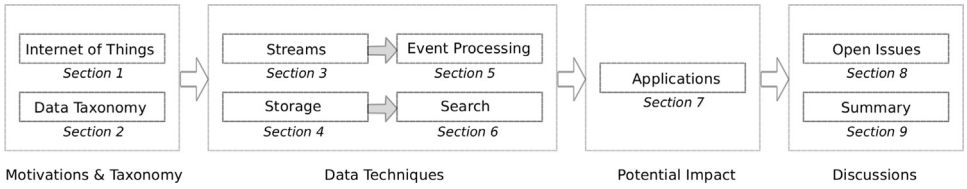

Fig. 2. Roadmap of this paper.

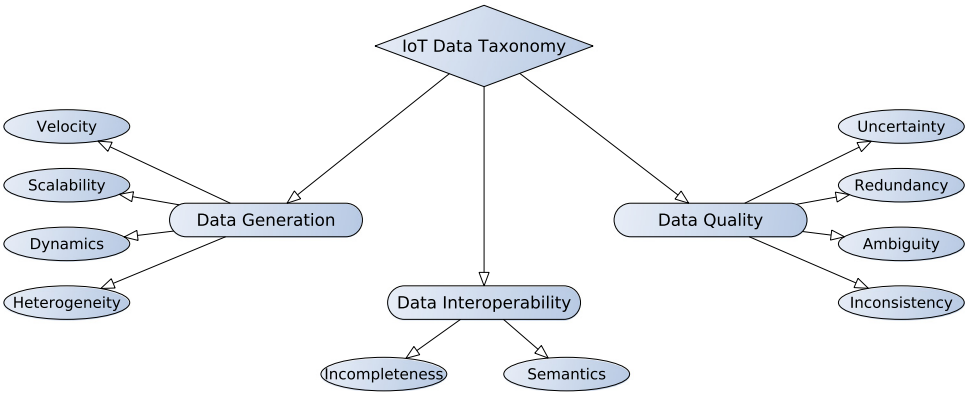

Fig. 3. IoT data taxonomy.

any other kinds of things, which brings about the challenges of proper interpretation of the produced data to different data consumers.

- *Inconsistency*: Inconsistency is also prevalent in IoT data. For example, in RFID data, inconsistency can occur due to missing readings of tags at some locations along the supply chain. It is also easy to observe inconsistency in sensing data as when multiple sensors are monitoring the same environment and reporting sensing results. Due to the precision and accuracy of the sensing process and other problems including packet loss during transmission, data inconsistency is also an intrinsic characteristic in sensing data.

2.3. Data interoperability

- *Incompleteness*: In order to process IoT data, being able to detect and react to events in real-time, it is important to combine data from different types of data sources to build a big and complete picture of relevant backgrounds of the real world. However, as this process relies on the cooperation of mobile and distributed things who are generating relevant background data, incompleteness is easily observable in IoT data. Suppose there are a large number of available data sources, it is of great importance to determine which data sources can best address the incompleteness of data for a given data processing task.
- *Semantics*: To address the challenges posed by the deluge of IoT data, things, or machines acting as the major data consumers should be a promising trend for data processing in the IoT era. Inspired by Semantic Web technologies, in order to enable machines to understand data for human beings, injecting semantics into data could be an initial step. Therefore, semantics within IoT data will play an important role in the process of enabling things/machines to understand and process IoT data by themselves.

3. Data streams

A data stream is a sequence of data objects of which the number is potentially *unbounded*. A data stream may be continuously generated at a rapid rate. In the data stream, each data object can be described by a multi-dimensional attribute vector within a continuous, categorical, or mixed attribute space (de Andrade Silva et al., 2013). There are some typical characteristics of data streams:

- Continuous arrival of data objects.
- Disordered arrival of data objects.
- Potentially unbounded size of a stream.
- Normally no persistence of data objects after being processed.
- Changing probability distributions of the unknown data generation process.

Due to the excessive amount of data produced by all kinds of things in the era of IoT, data streams play an important role in data processing and analysis. This section will focus on related data stream research efforts that can help handle IoT data. Our discussions include general data stream processing, RFID data stream processing, and RDF triple stream processing.

3.1. General data stream processing

Data streams can be generated in various scenarios, including a network of sensor nodes, a stock market or a network monitoring system. In many scenarios such as the sensor network scenario, sensor nodes are normally powered by batteries or solar panels. Therefore, in a typical sensor data processing system, one of the challenging issues is power constraints. In most applications, communication across sensor networks or with a centralized server requires the largest amount of energy as sensing consumes less energy (Subramaniam and Gunopulos, 2007). If sensor nodes send their raw sensing data to a server without consideration of the amount of energy needed to communicate, the battery life of the sensor nodes could be drastically reduced. Consequently, sensor data processing techniques, including data aggregation, data compression, modeling and online querying, should be performed *on-site* or *in-network* to reduce communication cost (Subramaniam and Gunopulos, 2007). Furthermore, numerous demands on efficient data processing algorithms for sensor systems arise due to the limitations of computational power of sensor nodes as well as the existence of inaccuracy and bias in the sensor readings. In other scenarios, such as stock market and network monitoring systems, there also exist challenges in processing high-rate data streams.

3.1.1. Query processing

There are several important queries to be considered (Subramaniam and Gunopulos, 2007):

- *Aggregate queries*: Aggregate queries is an important class of queries in sensor systems, including `MIN`, `COUNT` and `AVG` operators. Various techniques have been proposed to efficiently process these aggregate operators in sensor systems, which can help to effectively reduce power consumption. Considering the properties of the aggregate functions, the in-network partial data could be preprocessed first, which can then be utilized to produce the final results for the issued queries.
- *Join queries*: An example of join queries is "*Return the objects that were detected in both regions R1 and R2*" (Subramaniam and Gunopulos, 2007). To evaluate the query, stream readings from the sensors in regions R1 and R2 should be joined first before we can determine whether an object was detected in the two designated regions. Join queries are useful in many applications, such as monitoring an environment where multiple sensors are deployed and tracking moving objects that are monitored by several types of sensors.
- *Top-k monitoring*: The general problem of monitoring top-*k* values from distributed data streams is investigated in Babcock and Olston (2003). A technique is proposed to ensure the validity of the most recently communicated top-*k* answers by maintaining some specified arithmetic constraints at the stream sources. User specified error tolerance is also considered in order to provide high-quality answers. This technique can help reduce the overall communication cost between different sources.
- *Continuous queries*: To monitor designated changes in an environment, sensors are typically required to answer queries in continuous manner. For instance, motion or sound sensors might be used to evaluate some continuous queries, such as "Turn lights off if no motion is detected in area A in the past 10 min". When the query constraints are satisfied, the action of turning lights off could be automatically triggered by these sensors. If there are more than one continuous query evaluated over the same sensor readings, the storage and computation can be optimized by exploiting the fact that the sources of the queries and their partial results could overlap (Subramaniam and Gunopulos, 2007).

In IoT, query processing over streaming data will need to focus more on IoT related aspects of the streaming data, such as uncertainty, ambiguity and inconsistency. Furthermore, it is also imperative to address issues related to velocity and heterogeneity. For example, how can one efficiently aggregate the information stream from more than one thing with a large amount of ambiguity and inconsistency; how can one perform accurate join queries over uncertain IoT data streams with ambiguous and incomplete information; how can one identify top-*k* values from millions of heterogeneous IoT data streams efficiently and effectively; how can one monitor changes based on continuous queries from a large number of dynamic, fast, heterogeneous and incomplete IoT data streams, etc. In addition, new types of queries may also need to be considered, such as source selection queries for overcoming data incompleteness.

3.1.2. Stream mining

Stream mining can extract useful rules/information from data streams. Some typical tasks for stream mining are listed in the following:

- *Clustering*: Clustering is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar to each other than to those in other groups (clusters). Clustering techniques for data streams typically continuously cluster objects on memory constrained devices with some time limitations. Due to these restrictions, there are some requirements to consider when designing algorithms for clustering data streams (Gama, 2010): (i) providing clustering results via fast and incremental processing of data objects; (ii) rapidly detecting new clusters or changes of existing clusters; (iii) scaling to the potentially unbounded number of objects in data streams; (iv) providing a model representation that is consistently compact regardless of the number of data objects; (v) rapidly detecting the presence of outliers and acting accordingly; and (vi) dealing with different data types, such as XML trees, DNA sequences, GPS temporal and spatial information.
- *Classification*: Classification uses prior knowledge to guide the partitioning process to construct a set of classifiers to represent the possible distribution of patterns (Wang and Liu, 2011). Basically, compared with clustering, classification is a supervised learning process whereas clustering is an unsupervised learning process. More formally, a typical classification algorithm can be defined as follows (Wang and Liu, 2011): given a predefined classifier and two sets of data, labeled data and unlabeled data, the labeled data is used to train the classifier and the unlabeled data can then be classified by the trained classifier.
- *Outlier and anomaly detection*. In outlier and anomaly detection, the main task is to find data points that are most different from the remaining points in a given data set. Most existing outlier detection algorithms are based on the distance between every pair of points. The points that are most distant from all other points will be marked as outliers (Knorr and Ng, 1998). To be more specific, an object O in a dataset T is a $DB(p,D)$-outlier (DB here refers to distance-based) if at least fraction p of the objects in T lies greater than distance D from O. This kind of algorithms suffers from the same performance issue as they all run in $O(n^2)$ time. Hence, it is difficult to extend such approaches to distributed streaming data sets because points in those data sets normally arrive at multiple distributed end-points and must be processed incrementally.
- *Frequent itemset mining*: Frequent itemset mining is to find sets of items or values that co-occur frequently, or in other words, to find co-occurrence relationships in a transactional data set. Here a transactional data set refers to a data set where a set of items appears together in some specified context. Given a predefined support s, the goal in frequent itemset mining is to find all subsets of items that occur at least s number of times, or in other words, that appear in at least s transactional data sets at hand. Frequent itemset mining is both CPU and I/O intensive. Therefore, it is costly to completely re-mine a dynamic data set, which will be a typical case in IoT.

In IoT, multiple data streams processing would be more preferable as data streams can be generated at anywhere around the world and can be accessed globally via the Internet if being made public. For example, SmartSantander[3] proposes a city-scale experimental research facility in support of typical applications and services for a smart city. Around 20,000 sensors have been deployed to provide a variety of services, such as static environmental monitoring, mobile environmental monitoring, parks and gardens irrigation, outdoor parking area management, guidance to free parking lots and traffic intensity monitoring. A large number of data streams have to be processed efficiently to provide real-time monitoring of a smart city. Furthermore, how to efficiently and effectively mine IoT data streams that are highly dynamic, heterogeneous, uncertain, ambiguous, inconsistent, and incomplete will also require a revisit of the existing streaming mining techniques.

[3] http://www.smartsantander.eu/

3.2. RFID data stream processing

In 2003, a nonprofit open forum called the Ubiquitous ID Center[4] was established. So far, more than 500 companies and organizations worldwide have contributed to it, publishing uID standards and industrial open standard specifications. uID standards are based on the uID architecture (Koshizuka and Sakamura, 2010), which identifies real-world entities via Radio-Frequency Identification (RFID) tags or barcodes, determines contextual information such as environmental parameters from networked sensors, and adapts information services according to the data it obtains.

RFID systems consist of radio frequency (RF) tags (also called transponders) and RF tag readers (also called transceivers). Readers may be able to both read data from and write data to a transponder. RFID is a promising electronic identification technology that enables real-time monitoring and tracking applications in a variety of domains. Object identification information is stored on an RFID tag. This could be an Electronic Product Code (EPC).[5] EPC is a unique item identification code, which normally contains information about the manufacturer, the type of item and the serial number of the item (the tag ID). Streams of RFID reading data, whose basic form is a triplet $< tag\_id; reader\_id; timestamp >$, raise new challenges since the data may be insufficient, incomplete, and voluminous (Sheng et al., 2008).

In the past decade, the Auto-ID Center, now which is called the Auto-ID Labs,[6] has attracted industrial interests from companies and government initiatives to advance new developments and interests in RFID technology. One of the important advances is the so-called "Networked RFID" (Roussos, 2008). Networked RFID aims at connecting isolated RFID systems and software via the Internet. The EPcglobal Network, initially designed by the Auto-ID Labs and then further developed by EPCglobal at GS1,[7] is one of the notable efforts for Networked RFID.

In the following, we review some major RFID data stream processing techniques and summarize them in Table 1.

3.2.1. RFID data cleaning (uncertainty and unreliability)

SMURF (Statistical sMoothing for Unreliable RFid data) (Jeffery et al., 2006) is the first declarative, adaptive smoothing filter for cleaning raw RFID data streams. Unlike conventional techniques which expose the smoothing window parameter to the application, SMURF adapts the window size automatically and continuously over the lifetime of the system based on observed readings.

Periods of dropped readings and periods when a tag has moved are difficult to distinguish, which poses some challenges for the design of SMURF. To overcome such difficulty, a statistical sampling-based approach is put forward in SMURF. The main motivation is that RFID data streams can be modeled as a random sample of the tags in a reader's detection range. This sample-based view of observed RFID readings enables SMURF to develop algorithms based on statistical sampling theory to adapt the window size effectively.

Basically, the false reads in RFID streams can be classified into two categories (Liao et al., 2011):

- *Missing-reads*: Though an RFID tag is located in the range of a reader, it might not be read at all, thereby leading to a false prediction that the tag is not present. This may be caused by the weakness of RF signal, shortage of power, shield of signal between the tag and the reader, and the collision between tags. This type of errors is also referred as false negatives.
- *Cross-reads*: When an RFID tag locates outside the range of a reader, but it might be captured by this reader which leads another false prediction that the object is present in the scope of this reader (sometimes called *ghost reading*). Cross-reads may be arisen by the reflection of metal items, the abrupt strength of RF, and the change of antenna directions. This type of errors is also called false positives.

SMURF cannot eliminate the cross-reads generated by physical factors. A kernel density-based probability cleaning method, called KLEAP, can be used to filter the cross-reads in RFID data streams (Liao et al., 2011). KLEAP considers cross-reads as outliers, thus, the determination of cross-reads is transformed into the issue of detecting outliers on data streams. The density-based methods often perform better than the distance-based one, so KLEAP applies the density-based methods to detect cross-reads. It detects the exact positions of tags over the RFID data streams through examining the kernel densities of each tag captured by multiple readers.

The knowledge on the map of the real world and on the motility characteristics (such as the maximum speed) of the monitored objects is exploited by Fazzinga et al. (2014). From this knowledge of the domain, constraints can be naturally derived on the connectivity between pairs of locations (direct unreachability constraints) and/or on the time needed for reaching a location starting from another one (traveling-time constraints). These constraints can be used to discard interpretations of the data corresponding to inconsistent trajectories. Then a graph is built in the following way: its nodes correspond to pairs $< location, timestamp >$ and inside the graph, paths from source to target nodes one-to-one correspond to the valid trajectories in real word. Each node or edge is assigned a probability obtained by revising the a priori probability of the corresponding pair $< location, timestamp >$, so that the overall probability of a source-to-target path is the conditioned probability of the corresponding trajectory. In this way, trajectories of RFID-monitored objects can be cleaned.

3.2.2. RFID data inference and compression

RFID data inference techniques are closely related to RFID data cleaning techniques because inference techniques will need to clean RFID data first and then they can infer to the high level information about the tagged objects, i.e., location and containment relationships. Since raw RFID data contains a large amount of redundancies, RFID data compression is also applied to reduce space requirements after inference results have been obtained. RFID data compression is a further step beyond inference, where compression is performed based on the results of inference to remove the redundant data.

Noisy, raw data streams from mobile RFID readers are considered and a probabilistic approach to translate these streams into clean, rich event streams with location information is employed by Tran et al. (2009). Their probabilistic model is built based on the mobility of the reader, object dynamics, and noisy readings. *Particle filtering* is used to infer clean information about object locations from raw streams captured from mobile RFID readers.

The aforementioned data cleaning and inference techniques focus on smoothing over time, where containment relationships are not considered. Containment refers to *inter-object* relationships, e.g., containment between objects, cases, and pallets. Containment queries can be useful for enforcing packaging and shipping regulations. Some examples of containment queries have been provided by Cao et al. (2011), such as "raise an alert if a flammable item is not packed in a fireproof case" or "verify that the food containing peanuts is never exposed to other food cases for more than an hour". They also observe that some known containment relations can be used to determine object locations by smoothing over these facts. For example, suppose that we can

[4] uID Center: www.uidcenter.org
[5] http://www.epc-rfid.info/
[6] http://www.autoidlabs.org
[7] http://www.gs1.org/

Table 1
Comparisons of RFID streaming techniques.

Approach	Vel	Sca	Dyn	Het	Unc	Red	Amb	Incs	Incp	Sem
SMURF (Jeffery et al., 2006)			√		√					
KLEAP (Liao et al., 2011)			√		√			√		
Mobility-Monitoring (Fazzinga et al., 2014)			√		√	√			√	√
Mobile-RFID-Data-Cleaning (Tran et al., 2009)			√		√	√			√	
RFID-Data-Cleaning (Containment) (Cao et al., 2011; Nie et al., 2012)			√		√	√			√	√

Note: Key elements that have been considered are Vel: Velocity; Red: Redundancy; Sca: Scalability; Amb: Ambiguity; Dyn: Dynamics; Incs: Inconsistency; Het: Heterogeneity; Incp: Incompleteness; Unc: Uncertainty; Sem: Semantics.

infer that a specific set of objects has been packed in the same container. According to such knowledge, if one object in the container is read, all of the other objects must be in the same place. However, the fact is that the containment relationships are not known in advance. Therefore, a graphic model is proposed to infer containment relationships and to detect changes in containment relationships (Cao et al., 2011; Nie et al., 2012).

3.3. RDF triple stream processing

Linked Data is a method for publishing structured data and interlink such data to make it more useful.[8] It builds upon standard Web technologies such as HTTP, RDF and URIs and extends these technologies to share information. Linked Data can be understandable by computers. Data from different sources can be connected and queried in the form of Linked Data. Basically, Linked Data refers to a set of best practices to be followed in order to publish and link data on the Web, using the following basic principles[9]:

- Use URIs as names for things.
- Use HTTP URIs so that people can look up those names.
- When someone looks up a URI, provide useful information, using appropriate standards (RDF, SPARQL).
- Include links to other URIs, so that more things can be discovered.

The concept of *Linked Stream Data* applies the Linked Data principles to streaming data, so that data streams can be published as part of the Web of Linked Data. Stream reasoning can provide the abstractions, foundations, methods and tools required to integrate data streams, the Semantic Web and reasoning systems. Substantial research efforts have been put forward, focusing on how to apply reasoning on streaming data, how to publish raw streaming data and connect them to the existing data sets on the Semantic Web, and how to extend the SPARQL query language to process streaming data (Zhang et al., 2012). These research efforts lay some foundations of semantic IoT technologies, facilitating machine-to-machine communication in IoT.

3.3.1. Linked stream processing and reasoning

Efforts to apply the linked data principles to stream (sensor) data have been initiated and this wealth of information could be easily included in the Linked Data cloud.[10]

There are three typical streaming RDF/SPARQLS engines, including Streaming SPARQL (Bolles et al., 2008), SPARQLStream (Calbimonte et al., 2010), C-SPARQL (Barbieri et al., 2010), and EP-SPARQL (Anicic et al., 2011). Each of these systems also proposes its own SPARQL extension for streaming data processing. In these studies, SPARQL has been extended to have sliding window operators for RDF stream processing.

For example, Streaming SPARQL extends SPARQL to support window operators. But it does not consider performance issues, specially when designing the data structures. Further, it does not consider the sharing of computing states for continuous execution. Another example is SPARQLStream, which aims at enabling ontology-based access to streaming data. It defines a SPARQLStream language, which can be translated into another relational stream language based on mapping rules.

C-SPARQL (Continuous SPARQL) (Barbieri et al., 2010) attempts to facilitate reasoning upon rapidly changing information. In C-SPARQL, continuous queries are divided into static and dynamic parts and streaming data is transformed into non-streaming data within a specified window in order to apply standard algebraic operations, such as aggregate functions like `COUNT`, `COUNT DISTINCT`, `MAX`, `MIN` and `AVG`. The static parts will be loaded into relations, and the continuous queries are executed by processing the stream data against these relations. Event Processing SPARQL (EP-SPARQL), a language to describe event processing and stream reasoning, can be translated to ETALIS (Anicic et al., 2011), a Prolog-based complex event processing framework. First, RDF-based data elements are transformed into logic facts, and then EP-SPARQL queries are translated into Prolog rules.

Different from the above approaches, CQELS (Phuoc et al., 2011) is a native streaming RDF/SPARQL system built from scratch. CQELS defines and implements a native processing model in the query engine. Its query execution framework can also dynamically adapt the query processor to changes in the input data. By using data encoding and caching of intermediate query results, CQELS reduces external disk access on large Linked Data collections. Some indexing techniques are also adopted to enable faster data access. Table 2 compares all these systems from various aspects.

3.3.2. Extracting RDF triples from unstructured data streams

Although the current Linked Open Data (LOD) cloud has tremendously grown over the last few years, it delivers mostly encyclopedic information (such as albums, places, and kings) and fails to provide up-to-date information (Gerber et al., 2013). Based on such observation, they develop RdfLiveNews, an approach that allows extracting RDF from unstructured (i.e., textual) data streams in a fashion similar to the live versions of the DBpedia[11] and LinkedGeoData[12] datasets. RdfLiveNews takes unstructured data streams as its input. It firstly removes duplicates in the streams. Then it uses the cleaned streams as a basis to extract patterns for relations between known resources. Next, the patterns will be clustered to labeled relations and finally will be used as a basis for generating RDF triples.

[8] www.en.wikipedia.org
[9] http://www.w3.org/DesignIssues/LinkedData.html
[10] http://linkeddata.org/
[11] http://live.dbpedia.org/sparql
[12] http://live.linkedgeodata.org/sparql

Table 2
Comparisons of linked stream processing and reasoning.

Approach	Native	Aggregation Support	Reasoning Support	SPARQL 1.1 Support
Streaming SPARQL (Bolles et al., 2008)	No	Limited	Limited	Limited
SPARQLStream (Calbimonte et al., 2010)	No	Limited	Limited	Limited
C-SPARQL (Barbieri et al., 2010)	No	Rich	Limited	Limited
EP-SPARQL (Anicic et al., 2011)	No	Limited	Rich	Limited
CQELS (Phuoc et al., 2011)	Yes	Limited	Limited	Limited

4. Data storage models

The nature of data produced by the Internet of Things calls for a revisit of data storage techniques, which will be further discussed in this section.

4.1. New architecture

Traditional Database Management Systems (DBMSs) employ record-oriented (i.e., a record is represented by a row in a relational table) storage systems. With this row store architecture, a single disk write is able to store a single record with multiple attributes to disk. Records writes and updates are normally of high performance in these systems. Therefore, a DBMS with a row store architecture can be called a *write-optimized* system. In contrast, some systems need to deal with ad hoc querying of large amounts of data, where read performance is of more importance. For such systems, *read-optimized* is the major design factor. Take data warehouses as an example. They represent one class of read-optimized system. In these read-optimized systems, a column-store architecture is a better choice. This is because in a column-store system, the values for each single column (or attribute) are stored contiguously, which can be easily optimized for high-performance querying.

C-Store, a column-store architecture that supports the standard relational logical data model, has been designed by Stonebraker et al. (2005). Compared with the traditional DMBS architecture, the major differences are: (i) data in C-Store is not physically stored using its related relational logical data model; and (ii) whereas most row stores implement physical tables directly and then add various indexes to speed access, C-Store implements only projections. Here, projections are sorted subsets of the attributes of a table. Furthermore, superior performance of column store based systems has been shown over the major RDBMS (relational DBMS) system (Stonebraker et al., 2007). It is experimentally demonstrated that specialized engines in the data warehouse, stream processing, text, and scientific database markets can speed up the querying performance by 1–2 orders of magnitude using the column-store architecture. They also suggest that the DBMS vendors (and the research community) should start from scratch and design novel systems for requirements to be fulfilled in the near future, rather than just adapting current systems for those new requirements.

4.2. Large-scale storage in distributed environments

Storage issues in large scale systems have arisen due to the arrival of the big data era. For example, users of websites such as Facebook, Ebay and Yahoo! usually demand fast response times. One solution for this is to replicate data across globally distributed datacenters. However, it is discovered that to replicate all data to all locations may waste huge amounts of resources since users from different locations may have different data consumption needs (Kadambi et al., 2011). For example, an European server may not need to maintain a replica of some rare accessed records in an Asian server. By exploiting such observations, Kadambi et al. (2011) propose a selective replica strategy which supports replica of tables in the Web databases at record level to alleviate the overly replicated issue. In the selective replica strategy, each replica location stores a full or partial copy of the replicated table depending on the data needs. Specifically, in each location, a given record is stored either as a full replica or as a stub. A full replica is a normal copy of the record and possibly some metadata for supporting the selective replica strategy while a stub contains only the record's primary key and metadata. In this way, since large-scale Web databases are selectively replicated on a record-by-record basis, bandwidth and disk costs can be saved.

Therefore, to meet the exceptional demands of data storage in IoT, developments of large-scale, distributed storage systems are of essential. There are three factors or requirements to be considered when designing a distributed storage system (Chen et al., 2014):

- *Consistency:* Consistency means to ensure that multiple copies of the same data are identical since server failures and parallel storage may cause inconsistency.
- *Availability:* Availability refers to the requirement that the entire distributed storage system (which contains multiple servers) should not be seriously affected by some extent of server failures and should be able to provide satisfactory reading and writing performance.
- *Partition tolerance:* Since multiple servers are interconnected by a network and the data is partitioned across the network, the distributed storage system should have a certain level of tolerance to problems caused by network failures. This refers to partition tolerance requirement.

Interestingly, it has been proven by Gilbert and Lynch (2002) that a distributed storage system could not simultaneously meet the requirements on consistency, availability, and partition tolerance, and at most *two of the three requirements* can be satisfied at the same time. On top of this theory, there are three types of distributed storage systems: (1) a CA system, which ignores partition tolerance; (2) a CP system, which ignores availability; and (3) an AP system, which ignores consistency. The comparisons of these systems and some of their representative works are summarized in Table 3.

4.3. Storage on resource-constrained devices

Storage issues also arise in resource-constrained scenarios in IoT. For example, in sensor networks, communication activity normally plays a more important role than storage. But it is argued that for batch data collection, delay-tolerant mobile applications, and disconnected operations in static networks, the storage-centric paradigm becomes more critical (Mottola, 2010). It is favored by decreasing costs and increasing capacity of storage hardware. SQUIRREL is also proposed in the same work, which is a lightweight run-time layer allocating data to different storage areas, based on data size versus energy trade-offs.

SolarStore, a power storage service for solar-powered storage-centric sensor networks, has been developed by Yang et al. (2009). The main goal of SolarStore is to improve the total amount of data that can be eventually retrieved from the network. It adaptively balances data reliability against data sensing since solar energy is renewable and dynamic. For example, it chooses to replicate data in the network until the next opportunity to upload data to the server. The degree of data replication also varies dynamically depending on the availability of solar energy and sensor storage.

Table 3
Comparisons of three types of distributed storage systems.

Type	Pros	Cons	Representatives
CA	Single copy of data; consistency is easily ensured; availability is assured by the excellent design of databases	Could not handle network failures	Traditional small-scale relational databases
CP	Maintain several copies of the same data; a certain level of fault tolerance is ensured; consistency is ensured by guaranteeing multiple copies of data to be identical	Could not ensure sound availability due to the high cost for consistency assurance	BigTable Chang et al. (2008); Hbase Apache (2014)
AP	Maintain several copies of the same data; a certain level of fault tolerance is ensured; availability is assured by the design of distributed storage systems	Strong consistency is not ensured; May cause a certain amount of data errors	Dynamo DeCandia et al. (2007); Cassandra Lakshman and Malik (2010)

Table 4
Comparisons of search techniques.

Approach	Vel	Sca	Dyn	Het	Unc	Red	Amb	Incs	Incp	Sem
Twitter-Sensing (Sakaki et al., 2010)	√		√		√					
Real-time-Micro-Blogging-Search (Dong et al., 2010)	√		√		√				√	
MIDAS-RDF (Tsatsanifos et al., 2011)	√	√								
Web-Search-from-Structured-Databases (Agrawal et al., 2009)		√		√			√		√	√
Similarity-based-Entity-Search (Chaudhuri et al., 2009)		√		√			√		√	√
Snoogle (Wang et al., 2010)	√	√								
MAX (Yap et al., 2008)	√	√								
Microsearch (Tan et al., 2010)	√									
Dyser (Elahi et al., 2009)	√				√				√	
Collaborative-mobile-object-sensing (Frank et al., 2008)	√	√			√					

Note: Key elements that have been considered are Vel: Velocity; Red: Redundancy; Sca: Scalability; Amb: Ambiguity; Dyn: Dynamics; Incs: Inconsistency; Het: Heterogeneity; Incp: Incompleteness; Unc: Uncertainty; Sem: Semantics.

Early database systems for sensor networks such as TinyDB and Cougar only act as filters for data collection networks and not as databases, i.e., no data is stored in, or retrieved from, any database. A database management system for resource-constrained sensors named Antelope is presented by Tsiftes and Dunkels (2011). Antelope supports run-time creation and deletion of databases and indexes and hence is a dynamic database system. It is the first DBMS for resource-constrained sensor devices, which enables a class of sensor network systems where every sensor holds a database. It is envisioned that database techniques would become increasingly important in the progress of sensor network applications and energy-efficient storage. Further, indexing and querying would play important roles in emerging storage-centric applications.

Besides, flash storage has been used for logging data on a sensor node, which is called *amnesic storage* systems (Nath, 2009). An amnesic storage system archives streaming data using two key techniques: (i) data is compressed (usually with lossy compression methods) in an online fashion before being archived; and (ii) an amnesic storage system uses aging archived data by reducing the fidelity of older data to make space for newer data.

5. Search techniques

Searching and finding relevant objects from billions of things is one of the major challenges for the future Internet of Things and can bring about huge potential impact to humans. Supporting technologies for searching things in the IoT are very different from those used in searching Web documents because things are tightly bound to contextual information (e.g., location) and have no easily indexable properties (e.g., human readable text in the case of Web documents). In addition, the state information of things is dynamic and rapidly changing. Things discovery calls for innovative ways of managing and searching from dynamic data, which makes it different from traditional Web searching. This section overviews the relevant areas such as the Deep Web, Semantic Web and then discusses state-of-the-art techniques in searching things in the IoT environments. We also summarize these techniques in Table 4.

5.1. Deep Web and Semantic Web

Deep Web refers to the portion of content on the World Wide Web that is not indexed by standard search engines. Deep Web data is not directly being seen from Web pages but is accessible typically via HTML forms from the Web pages. The size of the Deep Web is estimated to be several orders of magnitude larger than that of the so-called *Surface Web* (the Web that is accessible and indexable by text search engines).

The Deep Web provides a wealth of hidden data in semi-structured form, accessible through Web forms and Web services. Since the data is hidden, to reach the whole content of the World Wide Web by just following hyperlinks is impossible. Regarding such issues, on top of XML, the Semantic Web grows as a common structured data source. With the W3C standards Resource Description Framework (RDF) and Web Ontology Language (OWL), the Semantic Web aims to unify the way semantic information is stored and exchanged. The Semantic Web makes it possible for machines themselves to not just read, but also "understand" the data from data sources, which enables machine to machine communication. In particular, languages such as Microformats[13] and schema.org can be used to add semantics to the descriptions of Web resources (including things).

5.2. Web search

The frequent changes and the unprecedented scale of the Web together pose enormous challenges to Web search engines, making it challenging to provide the most up-to-date and highly relevant information to its users. In IoT, this may become even more challenging as things would scale up the Web further and make the Web change more rapidly. For example, Tsubuyaku Sensor[14] is a

[13] http://www.microformats.org
[14] http://ts.uctec.com/tsensor/index-e.php

new wireless device from Japanese Ubiquitous Computing Technology. It can monitor conditions such as temperature, humidity and radiation levels. It then automatically tweets the resulting data via Twitter. In this way, a sensor becomes a virtual Twitter user, which can actively post tweets on the Web.

5.2.1. Real-time web search

Real-time web search refers to the retrieval of very latest content which is in high demand. It is reported that Twitter handled more than 50 million tweets per day in 2010 (Chen et al., 2011). Providing real-time search service is indeed very challenging in such large-scale microblogging systems because thousands of new updates need to be processed per second.

Twitter is real-time micro-blogging and the real-time interaction of events such as earthquakes in Twitter is investigated in Sakaki et al. (2010). They consider each Twitter user as a virtual sensor and apply Kalman filtering and particle filtering for estimating the centers of earthquakes and the trajectories of typhoons. Similarly, two challenges not encountered in non-real-time web search when supporting real-time web search have also been identified by Dong et al. (2010), which are (i) quickly crawling relevant content and (ii) ranking documents with link and click information. Then they propose to use the micro-blogging data stream to detect fresh URLs and to compute novel and effective features for ranking fresh URLs based on micro-blogging data.

5.2.2. Searching information over RDF data

Searching information from RDF data is important as more and more information is published in the form of RDF (e.g., via Linked Open Data Cloud). Efficient management of RDF data is also an important factor in realizing the Semantic Web vision. Performance and scalability issues need to be addressed as the Semantic Web technology is applied to real-world applications. Unlike the relational database community, the Semantic Web community uses a very different data model, which is RDF.

MIDAS-RDF, a distributed P2P RDF/S repository that is built on top of a distributed multi-dimensional index structure, has been presented by Tsatsanifos et al. (2011). It features fast retrieval of RDF triples satisfying various pattern queries by translating them into multi-dimensional range queries, which can be processed by the underlying index in hops logarithmic to the number of peers.

5.2.3. Collaborative web search

Web search engines often answer user queries based on data and information in relevant structured databases, which will be searched in isolation. Since a single database may not contain sufficient information to answer the query, the search often produces empty or incomplete results. Motivated by this observation, web search results and the items in structured databases have been exploited together to produce more complete answers to a wide range of queries that traditional web search cannot support well (Agrawal et al., 2009). Take query "light-weight gaming laptop" as an example. Dell XPS M1330 should be considered a match to such query as it is a light-weight laptop and suitable for gaming. But if searching only for the query keywords {`light-weight`, `gaming`} on the Web, Dell XPS M1330 may not appear in the search results. Therefore, the web search results (e.g., a set of relevant web documents) can be utilized to help identify relevant information in some structured databases (Agrawal et al., 2009). Then the user queries could be better answered.

Similarly, web search engines have also been exploited to define new similarity functions for recognizing named entities such as products, people names, or locations from documents, such as "X61" and the entity "Lenovo ThinkPad X61 Notebook" (Chaudhuri et al., 2009). The proposed new similarity functions are more accurate than existing string-based similarity functions because they aggregate evidence from multiple documents, and exploit web search engines to measure similarity.

5.3. Search of things in IoT

In IoT, connecting things enabled by RFID, embedded sensors and sensor networks to the Internet and publishing their output on the Web would become a reality. Real-world objects would have their own Web presence. Considering the potential and profound impact of IoT technologies, search of things in IoT will become as important as today's document search on the Web.

5.3.1. Keywords based search of things

Unlike search engines such as Google, searching for information in the physical world is more difficult because the physical objects do not have (reliable) connections to virtual space. For example, online books can be easily discovered by searching but physical books at home may be more difficult to find. This observation motivates the design of Snoogle (Wang et al., 2010), a search engine for the physical world. The basic idea behind Snoogle is that sensor nodes carry a textual description of the object they will be attached to. Such description forms the keywords for search of things. Then the keywords information of the whole sensor network is indexed using a two-tiered hierarchy. The lower tier contains many mediators, which are also called *index points*. Each index point maintains an aggregate view of all sensors in a local area (e.g., a room) and every sensor in the same area will be assigned to the same index point. In the top tier, there is a single mediator called the *key index point*. The key index point will maintain an aggregate view of the whole network.

MAX, a system that users can easily locate objects, is also designed (Yap et al., 2008). The main assumption is that tags are attached to everyday objects and each tag stores a descriptor of the object it is attached to (e.g., the book of `Harry Potter`). Multiple descriptor words are allowed in each tag, enabling users to label the object with richer information, so that others can locate the object based on the label. A three-tiered hierarchy of mediators is used. In the lowest tier, substations represent immobile objects such as tables or shelves, on which mobile tagged objects can be placed. In the middle tier, base stations represent a geographical space such as a room containing multiple substations. In the top tier, the MAX server represents the entire space covered by the system. When searching for an object left behind, it is easy to locate where the object has been left by exploiting the knowledge of which substation and base station it belongs to.

Microsearch is a system that runs on resource constrained small devices capable of being embedded into everyday objects (Tan et al., 2010). It allows users to do textual search in the local storage of a stand-alone small device, without support from a backend server. The challenge is that Microsearch runs in a resource constrained platform, where conventional search engine design and algorithms cannot be used. Information retrieval (IR) techniques for query resolution can answer top-k queries in a space-efficient manner (Tan et al., 2010).

Another search engine mainly designed for searching things, called Dyser, is proposed by Elahi et al. (2009). Dyser allows users to search for real-world entities with a given state, such as "hot" or "cold". However, this approach imposes two strong conditions: (i) to perform a query, end users have to know the vocabulary used by sensors (how states are named); and (ii) an entity must be represented by all the sensors that compose it. In order to estimate the probability of a sensor matching a query with sufficient accuracy and to rank sensor matching results, prediction models are adopted. The key idea of sensor ranking is to exploit the periodic nature of people-centric sensors by using appropriate prediction models.

Table 5
Comparisons of CEP Techniques.

Approach	Vel	Sca	Dyn	Het	Unc	Red	Amb	Incs	Incp	Sem
SASE (Jeffery et al., 2006)	√		√							
RFID-Streams-Pattern-Matching (Liao et al., 2011)	√		√							√
NEEL (Fazzinga et al., 2014)	√		√							√
FUGU (Tran et al., 2009)	√		√							
Resource-Constrained-CEP (Cao et al., 2011; Nie et al., 2012)	√		√		√				√	
KB-Fusion (Teymourian et al., 2012)									√	√
Semantic-Event-Enrichment (Hasan et al., 2013)									√	√
Approximate-Semantic-Matching (Zhou et al., 2011)					√					√
Heterogeneous-Approximate-Semantic-Matching (Hasan et al., 2012)				√	√					√

Note: Key elements that have been considered are Vel: Velocity; Red: Redundancy; Sca: Scalability; Amb: Ambiguity; Dyn: Dynamics; Incs: Inconsistency; Het: Heterogeneity; Incp: Incompleteness; Unc: Uncertainty; Sem: Semantics.

5.3.2. Collaborative search of things

A comprehensive system for managing and finding everyday objects relying on the collaboration of mobile phones in an urban area as object-sensing devices has been presented by Frank et al. (2008). For such tasks, the authors argue that the necessary infrastructures for such system include a sensing infrastructure, a communication infrastructure and a commercial infrastructure. Because of these requirements, the modern mobile phone system, which contains mobile sensors, provides a unique opportunity to realize collaborative search of everyday objects. The sensing model of the proposed system associates a probability with locations, meaning that the object currently has a certain probability of being at a certain location, thereby accelerating the search speed and reducing communication cost. Mobility provided by mobile sensors increases spatial coverage and hence the probability of finding a sought object.

6. Complex event processing

Data streaming techniques typically process incoming data through a sequence of transformations based on common SQL operators, like `selection`, `aggregate`, `join`, and these operators are defined in general by relational algebra. By contrast, the *complex event processing* (CEP) model views the information in the streams as events in the physical world. These events must be filtered, combined and transformed into higher-level events for better understanding by computers and humans. Similar to traditional publish–subscribe systems, CEP systems allow subscribers to express their interest in composite events. The focus of the CEP model is on detecting occurrences of particular patterns of (low-level) events indicating some higher-level events, which may be interesting to some particular event subscribers. In the era of IoT, CEP techniques lay part of the foundation of supporting computers to sense and react to events in the physical world. In the following, we survey some major CEP techniques related to IoT and summarize these techniques in Table 5.

6.1. Complex event processing

Systems for event processing and in particular event recognition (*event pattern matching*) accept a stream of time-stamped, simple or low-level events as input. A low-level event is the result of applying a computational derivation process to some other event, such as an event coming from a sensor. Using low-level events as input, (complex) event processing systems identify composite or high-level events of interest (Artikis and Paliouras, 2014). They are also collections of events that satisfy certain patterns.

SASE is a complex event processing system designed for monitoring queries over streams of RFID readings (Wu et al., 2006). The SASE defines its own declarative event language that combines filtering, correlation, and transformation of events. The overall structure of the SASE language contains the `EVENT` clause specifying event patterns, the `WHERE` clause specifying qualifications and the `WITHIN` clause specifying window sizes. To meet the needs of RFID-enabled monitoring applications, several operators are also defined, including the `ANY` operator, the `SEQ_` operator, the `SEQ_WITHOUT` operator, the `Selection` operator and the `WITHIN_` operator. In order to process SASE queries, a query plan in SASE adopts a subset of six operators: `sequence scan`, `sequence construction`, `selection`, `window`, `negation`, and `transformation`. Pipelined execution of the above operators is used. More specifically, if a query matches a current event and some previous events, these events will be emitted from sequence scan and sequence construction immediately and form an event sequence. This event sequence is then pipelined through the subsequence operators, and added to the final output. To realize sequence scan, the basis of the whole process, Non-deterministic Finite Automata (NFA) are used.

Pattern matching over streams has been studied by Agrawal et al. (2008). It presents two new challenges: (i) compared with languages for regular expression matching, languages for pattern matching over streams are significantly richer; and (ii) the conventional techniques for stream query processing are inadequate for efficient evaluation of pattern queries over streams. In order to represent each pattern query, a new query evaluation model is designed for processing pattern matching over RFID streams, employing a new type of automaton that comprises a nondeterministic finite automaton (NFA) and a match buffer, named NFA^b (Agrawal et al., 2008). Because of the powerful expressiveness of NFA, the semantics for the complete set of event pattern queries can be captured by the NFA^b model. Optimizations and query evaluation plans can also be produced and applied based on this model over event streams.

Nested CEP language called NEEL is proposed to support the flexible nesting of `AND`, `OR`, `Negation` and `SEQ` operators at any level (Liu et al., 2011). One NEEL query example is given in Fig. 4, which expresses “a critical condition that after being recycled and washed, a surgery tool is being put back into use without first being sharpened, disinfected and then checked for quality assurance” (Liu et al., 2011). Several techniques are also proposed to accelerate the evaluation of nested queries. Firstly, nested event expressions will be converted into normal forms by a normalization procedure. Secondly, a group of similar sub-expressions will be processed using prefix caching, suffix clustering methods and a customized physical execution strategy. Thirdly, an optimizer for optimal shared execution method is also designed based on the idea of iterative improvement. Compared with the traditional iterative nested execution, the optimized NEEL execution is up to two orders of magnitude faster.

Recent efforts have also been put on other aspects of complex event processing. For example, complex event processing in a distributed environment is studied and FUGU – an elastic allocator

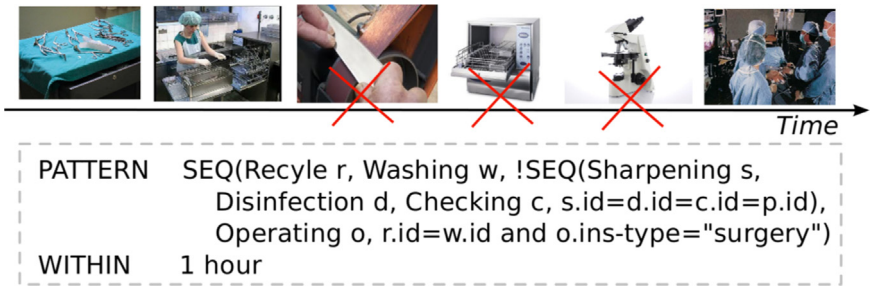

Fig. 4. Nested CEP query example (Adapted from the work by Liu et al., 2011).

for Complex Event Processing systems – is also proposed (Heinze et al., 2013). FUGU can dynamically allocate and de-allocate both stateless and stateful queries in order to meet the utilization goals. To that end, FUGU relies on bin packing to allocate queries to hosts. Very recently, load shedding techniques have been investigated for complex event processing under various resource constraints (He et al., 2014). Like other stream systems, CEP systems often face bursty input data. Since over-provisioning the system to the point where it can handle any such burst may be uneconomical or impossible, during peak loads a CEP system may need to "shed" portions of the load. The key technical challenge is to selectively shed work in order to eliminate the less important query results, thereby preserving the more useful query results defined by some utility function. Motivated by this, several load shedding algorithms are designed, including CPU-bound load shedding, memory-bound load shedding, and dual-bound load shedding (with both CPU- and memory-bound), depending on which resource is constrained.

6.2. Semantic complex event processing

The combination of event processing and knowledge representation can lead to novel semantic-rich event processing engines (Zhou et al., 2011; Teymourian et al., 2012). These intelligent event processing engines can (i) help to understand what is happening in terms of events, (ii) state and know what reactions and processes it can invoke, and furthermore (iii) decide what new events it can signal. The identification of critical events and situations requires processing vast amounts of data and meta-data within and outside the systems.

6.2.1. Semantic CEP system

A semantic CEP system is shown in Fig. 5. Semantic models of events can improve event processing quality by using event meta-data in combination with ontologies and rules (i.e., *knowledge bases*). The fusion of background knowledge with data from an event stream can help the event processing engine to know more about incoming events and their relationships to other related concepts. A Knowledge Base (KB) can be used to provide background knowledge about the events and other non-event resources (Teymourian et al., 2012). This means that events can be detected based on reasoning on their type hierarchy, temporal/spatial relationships, or their relationship to other objects in the application domain.

The benefits of using background knowledge in complex event processing can be seen as two major advantages over state-of-the-art CEP systems. The first benefit is its higher expressiveness and the second one its flexibility. Expressiveness means that an event processing system can precisely express complex event patterns and reactions to events which can be directly translated into business operations. Flexibility means that a CEP system is able to integrate new business changes into the systems in a fraction of time rather than changing the whole event processing rules. Complex event patterns are independent of current businesses and are defined in a higher level of abstraction based on business strategies. When something is changed in the business environment, it can be considered simply as an update in the background knowledge and the complex event detection patterns which are defined based on the business plans should not be changed.

6.2.2. Semantic event enrichment

The usage of background knowledge about events and their relations to other concepts in the application domain can improve the expressiveness and flexibility of CEP systems. Huge amounts of domain background knowledge stored in external knowledge bases can be used in combination with event processing in order to achieve more knowledgeable complex event processing.

An information completeness problem in semantic event processing contexts has been identified by Hasan et al. (2013) from a different angle. For example, while the basic information item in an event-based system is an event, normal users often require the system to handle information that is not encoded in the event. Such information typically comes from legacy databases or web data sources. This requires some degrees of information completeness or incompleteness for events to be sufficient for tasks such as subscription matching. The process of reducing information incompleteness is called *event enrichment*. Several challenges are identified for event enrichment, including determination of the enrichment source, retrieval of information items from the enrichment source, finding complementary information for an event in the enrichment source and fusion of complementary information with the event. To address these challenges, a model based on unifying enrichment within the event consumer logic and a native enricher that tackles incompleteness before matching are proposed (Hasan et al., 2013).

6.2.3. Approximate semantic matching

Approximate semantic matching is first studied by Zhou et al. (2011). To achieve approximate matching, semantic selection and inexact selection are used. More specifically, the semantic selection evaluates pattern constraints based on the semantic equivalence of attribute meanings captured by the event ontology instead of syntactic identical attribute values, while the inexact selection selects events and allows a limited number of mismatches to detect relevant patterns. A similarity function is associated with the inexact selection to evaluate relevance between matching patterns and target patterns.

Approximate semantic matching of heterogeneous events is also studied by Hasan et al. (2012). The motivation is that heterogeneous events are difficult to match in a distributed computing environment as similar or closely related events may not be described using the same words but in a semantically related form. To match all interesting events, users may have to write many slightly different subscriptions and have to know the exact format of all the heterogeneous events. Based on such observation, semantic decoupling of events and user's subscriptions becomes necessary. However, after such decoupling, the subscriptions would hardly exactly match

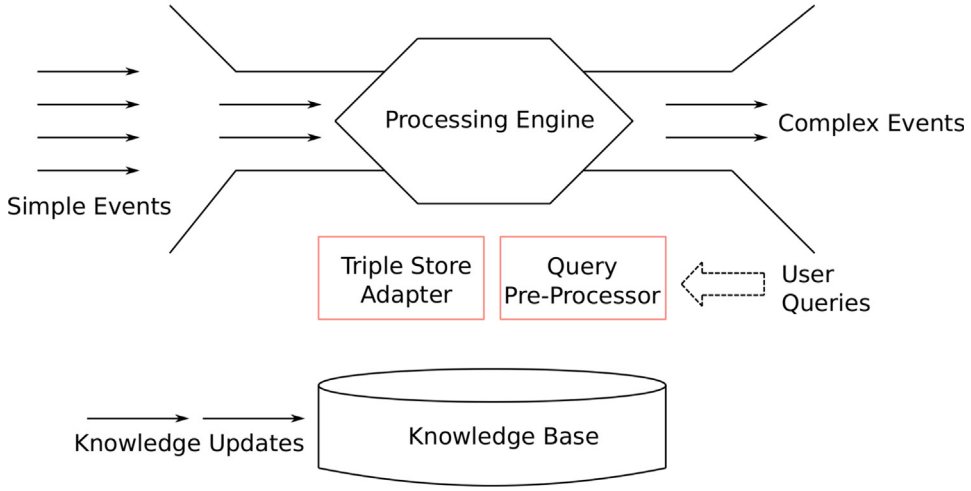

Fig. 5. Semantic complex event processing system overview (Adapted from the work by Teymourian et al., 2012).

the descriptions of events. This indicates that approximate matching and processing of events are inevitable. A model for approximate semantic matching that addresses event semantic decoupling is proposed. The model is evaluated using a hybrid matching approach based on both thesauri, semantic similarity and relatedness measures. After adopting this technique, the number of event subscriptions to achieve sufficiently precise matching results can be greatly reduced because of the decoupling between events and user subscriptions.

7. Potential IoT applications

As pointed out by Ashton (2009) that IoT "has the potential to change the world, just as the Internet did". The ongoing and/or potential IoT applications show that IoT can bring significant changes in many domains, i.e., cities and homes, environment monitoring, health, energy, and business. IoT can bring the ability to react to events in the physical world in an automatic, rapid and informed manner. This also opens up new opportunities for dealing with complex or critical situations and enables a wide variety of business processes to be optimized. In this section, we overview several representative domains where IoT can make some profound changes.

7.1. Smart cities and homes

IoT can connect billions of smart things and can help capture information in cities. Based on IoT, cities would become smarter and more efficient. Below are some examples of promising IoT applications in future smart cities. In a modern city, lots of digital data traces are generated there every second via cameras and sensors of all kinds (Guinard, 2010). All this data represents a goldmine for everyone, if people in the city would be able to take advantage of it in an efficient and effective way. For example, IoT can facilitate resources management issues for modern cities. Specifically, static resources (e.g., fire stations and parking spots) and mobile resources (e.g., police cars and fire trucks) in a city can be managed effectively using IoT technologies. Whenever events (fires, crime reports, cars looking for parking) arise, IoT technologies would be able to quickly match resources with events in an optimal way based on the information captured by smart things, thereby reducing cost and saving time. Taxi drivers in the city would also be able to better serve prospective passengers by learning passenger's mobility patterns and other taxi drivers' serving behaviors through the help of IoT technologies (Yuan et al., 2011). One study estimated a loss of $78 billion in 2007 in the form of 4.2 billion lost hours and 2.9 billion gallons of wasted gasoline in the United States alone (Mathur et al., 2010). IoT could bring fundamental changes in urban street-parking management, which would greatly benefit the whole society by reducing traffic congestion and fuel consumption.

Security in a city is of great concerns, which can benefit a lot from the development of IoT technologies. Losses resulted from property crimes were estimated to be $17.2 billion in the U.S. in 2008 (Guha et al., 2010). Current security cameras, motion detectors, and alarm systems are not able to help track or recover stolen property. IoT technologies can help to deter, detect, and track personal property theft since things are interconnected and can interact with each other. IoT technologies can also help improve stolen property recovery rates, and disrupt stolen property distribution networks. Similarly, a network of static and mobile sensors can be used to detect threats on city streets and in open areas such as parks.

With IoT technologies, people can browse and manage their homes via the Web. For example, they would be able to check whether the light in their bedrooms is on and could turn it off by simply clicking a button on a Web page. Similar operations and management could be done in office environments. Plumbing is ranked as one of the ten most frequently found problems in homes (Lai et al., 2010). It is important to determine the spatial topology of hidden water pipelines behind walls and underground. In IoT, smart things in homes would be able to report plumbing problems automatically and report to owners and/or plumbers for efficient maintenance and repair.

7.2. Environment monitoring

IoT technologies can also help to monitor and protect environments thereby improving human's knowledge about environments. Take water as an example. Understanding the dynamics of bodies of water and their impact on the global environment requires sensing information over the full volume of water. In such context, IoT technologies would be able to provide effective approaches to study water. Also IoT could improve water management in a city. Drinking water is becoming a scarce resource around the world. In big cities, efficiently distributing water is one of the major issues (Guinard, 2010). Various reports show that on average 30% of drinkable water is lost during transmission due to the aging infrastructure and pipe failures. Further, water can be contaminated biologically or chemically due to inefficient operation and management. In order to effectively manage and efficiently transport water, IoT technologies would be of great importance.

Soil contains vast ecosystems that play a key role in the Earth's water and nutrient cycles, but scientists cannot currently collect the high-resolution data required to fully understand them. Many soil

sensors are inherently fragile and often produce invalid or uncalibrated data (Ramanathan et al., 2009). IoT technologies would help to validate, calibrate, repair, or replace sensors, allowing to use available sensors without sacrificing data integrity and meanwhile minimizing the human resources required.

Sound is another example where IoT technologies can help. Sound is multidimensional, varying in intensity and spectra. So it is difficult to quantify, e.g., it is difficult to determine what kind of sound is noise. Further, the definitions and feelings of noise are quite subjective. For example, some noises could be pleasant, like flowing water, while others can be annoying, such as car alarms, screeching breaks and people arguing. A device has been designed and built to monitor residential noise pollution to address the above problems (Zimmerman and Robson, 2011). Firstly, noise samples from three representative houses are used, which span the spectrum of quiet to noisy neighborhoods. Secondly, a noise model is developed to characterize residential noise. Thirdly, noise events of an entire day (24 h) are compressed into a one minute auditory summary. Data collection, transmission and storage requirements can be minimized in order to utilize low-cost and low-power components, while sufficient measurement accuracy is still maintained.

Intel has developed smart sensors that can warn people about running outside when the air is polluted.[15] For example, if someone is preparing to take a jog along his/her regular route, an application on his/her smartphone pushes out a message: air pollution levels are high in the park where he/she usually runs. Then he/she could try a recommended route that is cleaner. Currently, many cities already have pollution and weather sensors. They are usually located on top of buildings, far from daily human activities.

7.3. Health

In future IoT environments, an RFID-enabled information infrastructure would be likely to revolutionize areas such as healthcare and pharmaceutical. For example, a healthcare environment such as a large hospital or aged care could tag all pieces of medical equipment (e.g., scalpels and thermometers) and drug products for inventory management. Each storage area or patient room would be equipped with RFID readers that could scan medical devices, drug products, and their associated cases. Such an RFID-based infrastructure could offer a hospital unprecedented near real-time ability to track and monitor objects and detect anomalies (e.g., misplaced objects) as they occur.

As personal health sensors become ubiquitous, they are expected to become interoperable. This means standardized sensors can wirelessly communicate their data to a device many people already carry today (e.g., mobile phones). It is argued by Lester et al. (2009) that one challenge in weight control is the difficulty of tracking food calories consumed and calories expended by activity. Then they present a system for automatic monitoring of calories consumed using a single body-worn accelerometer. To be fully benefited from such data for a large body of people, applying IoT technologies in such area would be a promising direction.

Mobile technology and sensors are creating ways to inexpensively and continuously monitor people's health. Doctors may call their clients to schedule an appointment, rather than vice versa, because the doctors could know their clients' health conditions in real-time. Some projects for such purpose have been initiated. For example, EveryHeartBeat[16] is a project for Body Computing to “connect the more than 5 billion mobile phones in the world to the health ecosystem”. In the initial stage, heart rate monitoring is investigated. Consumers would be able to self-track their pulse and studies show heart rate monitoring could be useful in detecting heart conditions and enabling early diagnosis. The future goal is to include data on blood sugar levels, and other biometrics collected via mobile devices.

7.4. Energy

Home heating is a major factor in worldwide energy use. In IoT, home energy management applications could be built upon embedded Web servers (Priyantha et al., 2008). Through such online web services, people can track and manage their home energy consumption. A system is designed by Gupta et al. (2009) for augmenting these thermostats using just-in-time heating and cooling based on travel-to-home distance obtained from location-aware mobile phones. The system makes use of a GPS-enabled thermostat which could lead to savings of as much as 7%. In IoT, as things in homes would become smart and connected to the Internet, similar energy savings could be more effective. For example, by automatically sensing occupancy and sleep patterns in a home, it would be possible to save energy by automatically turning off the home's HVAC (heating, ventilation, and air conditioning) system.

Besides home heating, fuel consumption is also an important issue. GreenGPS, a navigation service that uses participatory sensing data to map fuel consumption on city streets, has been designed by Ganti et al. (2010). GreenGPS would allow drivers to find the most fuel-efficient routes for their vehicles between arbitrary end-points. In IoT, fuel consumption would be further reduced by enabling cars and passengers to communicate with each other for ride sharing (Yuan et al., 2011).

7.5. Business

IoT technologies would be able to help to improve efficiency in business and bring other impacts on business (Mattern and Floerkemeier, 2010):

- From a commercial point of view, IoT can help increase the efficiency of business processes and reduce costs in warehouse logistics and in service industries. This is because more complete and necessary information can be collected by interconnected things. owing to its huge and profound impact on the society, IoT research and applications can also trigger new business models involving smart things and associated services.
- From a social and political point of view, IoT technologies can provide a general increase in the quality of life for the following reasons. Firstly, consumers and citizens will be able to obtain more comprehensive information. Secondly, care for aged and/or disabled people can be improved with smarter assistance systems. Thirdly, safety can be increased. For example, road safety can be improved by receiving more complete and real-time traffic and road condition information.
- From a personal point of view, new services enabled by IoT technologies can make life more pleasant, entertaining, independent and also safer. For example, business taking advantages of technologies of search of things in IoT can help locate lost things quickly, such as personal belongs, pets or even other people.

Besides, take improving information handover efficiency in a global supply chain as an example. The concept of *digital object memories* (DOM) is proposed in Stephan et al. (2010), which can store order-related data via smart labels on the item. Based on DOM, relevant life cycle information could be attached to the product itself. Considering the potential different stakeholders including manufacturer, distributor, retailer, and end customer along the supply/value chain, this approach facilitates information handover.

[15] http://www.fastcoexist.com/1680111/intels-sensors-will-warn-you-about-running-outside-when-the-air-is-polluted

[16] http://join.everyheartbeat.org/

Further, there are many important bits of information in an IoT-based supply chain, such as the 5W (what, when, where, who, which). It is also necessary to integrate them efficiently and in real-time in other operations. The EPCIS (Electronic Product Code Information System) network is a set of tools and standards for tracking and sharing RFID-tagged products in IoT. However, much of this data remains in closed networks and is hard to integrate (Wu et al., 2013). IoT technologies could be used to make it easier to use all this data, to integrate it into various applications, and to build more flexible, scalable, global application for better (even real-time) logistics.

8. Open issues

The development of IoT technologies and applications is merely beginning. Many new challenges and issues have not been addressed, which require substantial efforts from both academia and industry. In this section, we identify some key directions for future research and development from a data-centric perspective:

- *Data quality and uncertainty:* In IoT, as data volume increases, inconsistency and redundancy within data would become paramount issues. One of the central problems for data quality is *inconsistency detection* and when data is distributed, the detection would be far more challenging (Fan et al., 2010). This is because inconsistency detection often requires shipping data from one site to another. Meanwhile, inherited from RFID data and sensor data, IoT data would be of great uncertainty, which also presents significant challenges.
- *Co-space data:* In an IoT environment, the physical space and the virtual (data) space co-exist, and interact simultaneously. Novel technologies must be developed to allow data to be processed and manipulated seamlessly between the real and digital spaces (Ooi et al., 2009). To synchronize data in both real and virtual worlds, large amount of data and information will flow between co-spaces, which pose new challenges. For example, it would be challenging to process heterogeneous data streams in order to model and simulate real world events in the virtual world. Besides, more intelligent processing is needed to identify and send interesting events in the co-space to objects in the physical world.
- *Transaction handling:* When the data being updated is spread across hundreds or thousands of networked computers/smart things with differing update policies, it would be difficult to define what the transaction is. In addition, most of things are resource-constrained, which are typically connected to the Internet using light-weight, *stateless* protocols such as CoAP (Constrained Application Protocol)[17] and 6LoWPAN (IPv6 over Low Power Wireless Personal Area Networks)[18] and accessed using RESTful Web services. This makes transaction handling in IoT a great challenge. As pointed out by James et al. (2009) that the problem is that the world is changing fast, the data representing the world is on multiple networked computers/smart things and existing database technologies cannot manage. Techniques developed for streamed and real-time data may provide some hints.
- *Frequently Updated Timestamped Structured (FUTS) data:* The Internet, and hence IoT, contains potentially billions of Frequently Updated Timestamped Structured (FUTS) data sources, such as real-time traffic reports, air pollution detection, temperature monitoring, and crops monitoring. FUTS data sources contain states and updates of physical world things. Current technologies are not capable in dealing with FUTS data sources (James et al., 2009) because (i) no data management system can easily display FUTS past data; (ii) no efficient crawler or storage engine is able to collect and store FUTS data; and (iii) querying and delivering FUTS data is hardly supported. All these pose great challenges for the design of novel data management systems for FUTS data.
- *Distributed and mobile data:* In IoT, data will be increasingly distributed and mobile. Different from traditional mobile data, distributed and mobile data in IoT would be much more highly distributed and data intensive. In the context of interconnecting huge numbers of mobile and smart objects, centralized data stores would not be a suitable tool to manage all the dynamics of mobile data produced in IoT. Thus there is a need for novel ways to manage distributed and mobile data efficiently and effectively in IoT.
- *Semantic enrichment and semantic event processing:* The full potentials of IoT would heavily rely on the progress of semantic Web. This is because things and machines should play a much more important role than humans in IoT to process and understand data. This calls for new research in semantic technologies. For example, there are increasing efforts in building public knowledge bases (such as DBpedia, FreeBase, and Linked Open Data Cloud). But how these knowledge bases can be effectively used to add to the understanding of raw data coming from sensor data streams and other types of data streams? To resolve this challenge, semantic enrichment of sensing data is a promising research direction. Further, consider the potential excessively large amount of subscriptions to IoT data. To produce proper semantic enrichment to meet different enrichment needs from different subscribers poses great challenges. Finally, how to effectively incorporate semantic enrichment techniques with semantic event processing to provide much better expressiveness in event processing is still at its initial stage. This will also demand a large amount of research efforts.
- *Mining:* Data mining aims to facilitate the exploration and analysis of large amounts of data, which can help to extract useful information for huge volume of IoT data. Data mining challenges may include extraction of temporal characteristics from sensor data streams, event detection from multiple data streams, data stream classification, activity discovery and recognition from sensor data streams. Besides, clustering and table summarization in large data sets, mining large (data, information or social) networks, sampling, and information extraction from the Web are also great challenges in IoT.
- *Knowledge discovery:* Knowledge discovery is the process of extracting useful knowledge from data. This is essential especially when connected things populate their data to the Web. The following issues related to knowledge discovery in IoT have been identified by Weikum (2011): (i) automatic extraction of relational facts from natural-language text and multi-modal contexts; (ii) large-scale gathering of factual-knowledge candidates and their reconciliation into comprehensive knowledge bases; (iii) reasoning on uncertain hypotheses, for knowledge discovery and semantic search; and (iv) deep and real-time question answering, e.g., to enable computers to win quiz game shows.
- *Security:* Due to the proliferation of embedded devices in IoT, effective device security mechanisms are essential to the development of IoT technologies and applications. National Intelligence Council (Anonymous, 2008) argues that, to the extent that everyday objects become information security risks, the IoT could distribute those risks far more widely than the Internet has to date. For example, RFID security presents many challenges. Potential solutions should consider aspects from hardware and wireless protocol security to the management, regulation and sharing of collected RFID data (Welbourne et al., 2009). Besides, it is argued by Lagesse et al. (2009) that there is still no generic framework for deploying and extending traditional security mechanisms over a variety of pervasive systems. Regarding

[17] http://tools.ietf.org/html/draft-ietf-core-coap-18
[18] http://tools.ietf.org/wg/6lowpan

security concerns of the network layer, it is suggested by Kounavis et al. (2010) that the Internet can be gradually encrypted and authenticated based on the observations that the recent advances in implementation of cryptographic algorithms have made general purpose processors capable of encrypting packets at high rates. But how to generalize such algorithms to IoT would be challenging as things in IoT normally only maintain low transmission rates and connections are usually intermittent.

- *Privacy:* Privacy protection is a serious challenge in IoT. One of the fundamental problems is the lack of a mechanism to help people expose appropriate amounts of their identity information. Embedded sensing is becoming more and more prevalent on personal devices such as mobile phones and multi-media players. Since people are typically wearing and carrying devices capable of sensing, details such as activity, location, and environment could become available to other people. Hence, personal sensing can be used to detect their physical activities and bring about privacy concerns (Klasnja et al., 2009).
- *Social concerns:* Since IoT connects everyday objects to the Internet, social concerns would become a hot topic in the development of IoT. Further, online social networks with personal things information may incur social concerns as well, such as disclosures of personal activities and hobbies. Appropriate economic and legal conditions and a social consensus on how the new technical opportunities in IoT should be used also represents a substantial task for the future (Mattern and Floerkemeier, 2010).

9. Summary

It is predicted that the next generation of the Internet will be composed of trillions of connected computing nodes at a global scale. Through these nodes, everyday objects in the world can be identified, connected to the Internet and take decisions independently. In this context, Internet of Things (IoT) is considered a new revolution of the Internet. In IoT, the possibility of seamlessly merging the real and the virtual worlds, through the massive deployment of embedded devices, opens up many new and exciting directions for both research and development. In this paper, we have provided an overview of some key research areas of IoT, specifically from a data-centric perspective. It also presents a number of fundamental issues to be resolved before we can fully realize the promise of IoT applications. This paper covers investigations on data models, data storage, stream processing, search and event processing. The most relevant application fields have also been reviewed.

Over the last few years, the Internet of Things has gained momentum and is becoming a rapidly expanding area of research and business. Many efforts from researchers, vendors and governments have been devoted to creating and developing novel IoT applications. Along with the current research efforts, we encourage more insights into the problems of this promising technology, and more efforts in addressing the open research issues identified in this paper.

Acknowledgements

Quan Z. Sheng's work has been supported by the Australian Research Council Discovery Grant DP140100104. We express our gratitude to the anonymous reviewers for their comments and suggestions which have greatly helped us to improve this work.

References

Agrawal J, Diao Y, Gyllstrom D, Immerman N. Efficient pattern matching over event streams. In: Proceedings of the ACM SIGMOD international conference on management of data (SIGMOD'08); 2008. p. 147–60.
Agrawal S, Chakrabarti K, Chaudhuri S, Ganti V, König AC, Xin D. Exploiting web search engines to search structured databases. In: Proceedings of the 18th international conference on world wide web (WWW). Madrid, Spain: ACM; 2009. p. 501–10.
An J, Gui X, Zhang W, Jiang J, Yang J. Research on social relations cognitive model of mobile nodes in Internet of Things. J Netw Comput Appl 2013;36(2):799–810.
Anicic D, Fodor P, Rudolph S, Stojanovic N. EP-SPARQL: a unified language for event processing and stream reasoning. In: Proceedings of the 20th international conference on world wide web (WWW); 2011. p. 635–44.
Anonymous. National Intelligence Council (NIC). Disruptive civil technologies: six technologies with potential impacts on us interests out to 2025. Conference report CR, 2008–07,⟨http://www.dni.gov/nic/NIC_home.html⟩; April 2008.
Apache. Apache HBase project ⟨http://www.hbase.apache.org/⟩; 2014.
Artikis A, Paliouras G. Tutorial: formal methods for event processing. In: Proceedings of the 17th international conference on extending database technology (EDBT); 2014. p. 675.
Ashton K. 'That internet of things' thing ⟨http://www.rfidjournal.com/article/view/4986⟩; 2009.
Atzori L, Iera A, Morabito G. The internet of things: a survey. Comput Netw 2010;54(15):2787–805.
Babcock B, Olston C. Distributed top-k monitoring. In: Proceedings of the 2003 ACM SIGMOD international conference on management of data (SIGMOD conference); 2003. p. 28–39.
Barbieri DF, Braga D, Ceri S, Grossniklaus M. An execution environment for C-SPARQL queries. In: Proceedings of the 13th international conference on extending database technology (EDBT); 2010. p. 441–52.
Bolles A, Grawunder M, Jacobi J. Streaming SPARQL-Extending SPARQL to Process Data Streams. In: Proceedings of the fifth European semantic web conference on the semantic web: research and applications (ESWC); 2008. p. 448–62.
Calbimonte J-P, Corcho Ó, Gray AJG. Enabling Ontology-Based Access to Streaming Data Sources. In: Proceedings of the ninth international semantic web conference (ISWC); 2010. p. 96–111.
Cao Z, Sutton CA, Diao Y, Shenoy PJ. Distributed inference and query processing for RFID tracking and monitoring. Proc VLDB Endow 2011;4(5):326–37.
CASAGRAS. CASAGRAS (Coordination and support action for global rfid-related activities and standardisation); 2000.
Chang F, Dean J, Ghemawat S, Hsieh WC, Wallach DA, Burrows M, et al. Bigtable: a distributed storage system for structured data. ACM Trans Comput Syst 2008;26:2.
Chaudhuri S, Ganti V, Xin D. Exploiting web search to generate synonyms for entities. In: Proceedings of the 18th international conference on world wide web (WWW). Madrid, Spain: ACM; 2009. p. 151–60.
Chen C, Li F, Ooi BC, Wu S. TI: an efficient indexing mechanism for real-time search on tweets. In: Proceedings of the ACM SIGMOD international conference on management of data (SIGMOD). Athens, Greece: ACM; 2011. p. 649–60.
Chen M, Mao S, Liu Y. Big data: a survey. Mob Netw Appl J 2014;19(2):171–209.
de Andrade Silva J, Faria ER, Barros RC, Hruschka ER, de Carvalho ACPLF, Gama J. Data stream clustering: a survey. ACM Comput Surv 2013;46(1):13.
DeCandia G, Hastorun D, Jampani M, Kakulapati G, Lakshman A, Pilchin A, et al. Dynamo: amazon's highly available key-value store. In: Proceedings of the 21st ACM symposium on operating systems principles (SOSP); 2007. p. 205–20.
Dong A, Zhang R, Kolari P, Bai J, Diaz F, Chang Y, et al. Time is of the Essence: improving recency ranking using twitter data. In: Proceedings of the 19th international conference on world wide web (WWW). Raleigh, NC, USA: ACM; 2010. p. 331–40.
Elahi BM, Römer K, Ostermaier B, Fahrmair M, Kellerer W. Sensor ranking: a primitive for efficient content-based sensor search. In: Proceedings of the eighth international conference on information processing in sensor networks (IPSN). San Francisco, CA, USA: IEEE; 2009. p. 217–28.
European Commission. European Commission: internet of things, an action plan for Europe ⟨http://europa.eu/legislation_summaries/information_society/internet/si0009_en.htm⟩; 2009.
Fan W, Geerts F, Ma S, Müller H. Detecting Inconsistencies in Distributed Data. In: Proceedings of the 26th international conference on data engineering (ICDE). Long Beach, CA, USA: IEEE; 2010. p. 64–75.
Fazzinga B, Flesca S, Furfaro F, Parisi F. Cleaning trajectory data of RFID-monitored objects through conditioning under integrity constraints. In: Proceedings of the 17th international conference on extending database technology (EDBT); 2014. p. 379–90.
Frank C, Bolliger P, Mattern F, Kellerer W. The sensor internet at work: locating everyday items using mobile phones. Pervas Mob Comput 2008;4(3):421–47.
Gama J. Knowledge discovery from data streams. Chapman and Hall/CRC data mining and knowledge discovery series. Boca Raton, FL: CRC Press; 2010.
Ganti RK, Pham N, Ahmadi H, Nangia S, Abdelzaher TF. GreenGPS: a participatory sensing fuel-efficient maps application. In: Proceedings of the eighth international conference on mobile systems, applications, and services (MobiSys). San Francisco, CA, USA: ACM; 2010. p. 151–64.
Gerber D, Hellmann S, Bühmann L, Soru T, Usbeck R, Ngomo A-CN. Real-time RDF extraction from unstructured data streams. In: Proceedings of the 12th international semantic web conference (ISWC); 2013. p. 135–50.

Gilbert S, Lynch NA. Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. SIGACT News 2002;33(2):51–9.
Guha S, Plarre K, Lissner D, Mitra S, Krishna B, Dutta P, et al. AutoWitness: locating and tracking stolen property while tolerating gps and radio outages. In: Proceedings of the eighth international conference on embedded networked sensor systems (SenSys). Zurich, Switzerland: ACM; 2010. p. 29–42.
Guinard D. A web of things for smarter cities. In: Technical talk; 2010. p. 1–8.
Guo B, Zhang D, Wang Z, Yu Z, Zhou X. Opportunistic IoT: exploring the harmonious interaction between human and the internet of things. J Netw Comput Appl 2013;36(6):1531–9.
Gupta M, Intille SS, Larson K. Adding GPS-control to traditional thermostats: an exploration of potential energy savings and design challenges. In: Proceedings of the seventh international conference on pervasive computing (Pervasive). Nara, Japan: Springer; 2009. p. 95–114.
Hasan S, O'Riain S, Curry E. Approximate semantic matching of heterogeneous events. In: Proceedings of the sixth ACM international conference on distributed event-based systems (DEBS); 2012. p. 252–63.
Hasan S, O'Riain S, Curry E. Towards unified and native enrichment in event processing systems. In: Proceedings of the seventh ACM international conference on distributed event-based systems (DEBS); 2013. p. 171–82.
He Y, Barman S, Naughton JF. On load shedding in complex event processing. In: Proceedings of the 17th international conference on database theory (ICDT); 2014. p. 213–24.
Heinze T, Ji Y, Pan Y, Grüneberger FJ, Jerzak Z, Fetzer C. Elastic complex event processing under varying query load. In: Proceedings of the first international workshop on big dynamic distributed data; 2013. p. 25–30.
INFSO. INFSO D.4 Networked enterprise & RFID INFSO G.2 micro & nanosystems. In: Co-operation with the working group RFID of the ETP EPOSS, Internet of things in 2020, Roadmap for the future, version 1.1; 27 May 2008.
ITU. International telecommunication union (ITU) internet reports. The internet of things; November 2005.
James AE, Cooper J, Jeffery KG, Saake G. Research directions in database architectures for the internet of things: a communication of the first international workshop on database architectures for the internet of things (DAIT 2009). In: Proceedings of the 26th British national conference on databases (BNCOD). Birmingham, UK: Springer; 2009. p. 225–33.
Jeffery SR, Garofalakis MN, Franklin MJ. Adaptive cleaning for RFID data streams. In: Proceedings of the 32nd international conference on very large data bases (VLDB); 2006. p. 163–74.
Kadambi S, Chen J, Cooper BF, Lomax D, Ramakrishnan R, Silberstein A, et al. Where in the world is my data? Proc VLDB Endow 2011;4(11):1040–50.
Klasnja PV, Consolvo S, Choudhury T, Beckwith R, Hightower J. Exploring privacy concerns about personal sensing. In: Proceedings of the seventh international conference on pervasive computing (Pervasive). Nara, Japan: Springer; 2009. p. 176–83.
Knorr EM, Ng RT. Algorithms for mining distance-based outliers in large datasets. In: Proceedings of 24th international conference on very large data bases (VLDB); 1998. p. 392–403.
Koshizuka N, Sakamura K. Ubiquitous ID: standards for ubiquitous computing and the internet of things. IEEE Pervas Comput 2010;9(4):98–101.
Kounavis ME, Kang X, Grewal K, Eszenyi M, Gueron S, Durham D. Encrypting the internet. In: Proceedings of the ACM SIGCOMM conference on applications, technologies, architectures, and protocols for computer communications (SIGCOMM). New Delhi, India: ACM; 2010. p. 135–46.
Lagesse B, Kumar M, Paluska JM, Wright M. DTT: a distributed trust toolkit for pervasive systems. In: Proceedings of the seventh annual IEEE international conference on pervasive computing and communications (PerCom). Galveston, TX, USA: IEEE; 2009. p. 1–8.
Lai T-T, Chen Y-H, Huang P, Chu H-H. PipeProbe: a mobile sensor droplet for mapping hidden pipeline. In: Proceedings of the eighth international conference on embedded networked sensor systems (SenSys). Zurich, Switzerland: ACM; 2010. p. 113–26.
Lakshman A, Malik P. Cassandra: a decentralized structured storage system. Oper Syst Rev 2010;44(2):35–40.
Lester J, Hartung C, Pina L, Libby R, Borriello G, Duncan G. Validated caloric expenditure estimation using a single body-worn sensor. In: Proceedings of the 11th international conference on ubiquitous computing (Ubicomp). Orlando, FL, USA: ACM; 2009. p. 225–34.
Li S, Xu LD, Zhao S. The Internet of things: a survey. Inf Syst Front 17 (2), 2015, 243-259.
Liao G, Li J, Chen L, Wan C. KLEAP: an efficient cleaning method to remove cross-reads in RFID streams. In: Proceedings of the 20th ACM conference on information and knowledge management (CIKM); 2011. p. 2209–12.
Liu M, Rundensteiner EA, Dougherty DJ, Gupta C, Wang S, Ari I, et al. High-performance nested CEP query processing over event streams. In: Proceedings of the 27th international conference on data engineering (ICDE); 2011. p. 123–34.
Mathur S, Jin T, Kasturirangan N, Chandrasekaran J, Xue W, Gruteser M, et al. ParkNet: drive-by sensing of road-side parking statistics. In: Proceedings of the eighth international conference on mobile systems, applications, and services (MobiSys). San Francisco, CA, USA: ACM; 2010. p. 123–36.
Mattern F, Floerkemeier C. From the internet of computers to the internet of things. In: From active data management to event-based systems and more. Berlin, Heidelberg: Springer; 2010. p. 242–59.
Mottola L. Programming storage-centric sensor networks with squirrel. In: Proceedings of the ninth international conference on information processing in sensor networks (IPSN). Stockholm, Sweden: IEEE; 2010. p. 1–12.
Nath S. Energy efficient sensor data logging with amnesic flash storage. In: Proceedings of the eighth international conference on information processing in sensor networks (IPSN). San Francisco, CA, USA: IEEE; 2009. p. 157–68.
Nie Y, Cocci R, Cao Z, Diao Y, Shenoy PJ. SPIRE: efficient data inference and compression over rfid streams. IEEE Trans Knowl Data Eng 2012;24(1):141–55.
Ooi BC, Tan K-L, Tung AKH. Sense the physical, walkthrough the virtual, manage the Co (existing) spaces: a database perspective. SIGMOD Rec 2009;38(3):5–10.
Perera C, Zaslavsky AB, Christen P, Georgakopoulos D. Context aware computing for the internet of things: a survey. CoRR abs/1305.0982, 2013.
Phuoc DL, Dao-Tran M, Parreira JX, Hauswirth M. A native and adaptive approach for unified processing of linked streams and linked data. In: Proceedings of the 10th international semantic web conference (ISWC); 2011. p. 370–88.
Priyantha NB, Kansal A, Goraczko M, Zhao F. Tiny Web Services: design and implementation of interoperable and evolvable sensor networks. In: Proceedings of the sixth international conference on embedded networked sensor systems (SenSys). Raleigh, NC, USA: ACM; 2008. p. 253–66.
Ramanathan N, Schoellhammer T, Kohler E, Whitehouse K, Harmon T, Estrin D. Suelo: human-assisted sensing for exploratory soil monitoring studies. In: Proceedings of the seventh international conference on embedded networked sensor systems (SenSys). Berkeley, CA, USA: ACM; 2009. p. 197–210.
Roussos G. Networked RFID—systems, software and services; 2008. p. 1–168.
Sakaki T, Okazaki M, Matsuo Y. Earthquake shakes twitter users: real-time event detection by social sensors. In: Proceedings of the 19th international conference on world wide web (WWW). Raleigh, NC, USA: ACM; 2010. p. 851–60.
Sheng QZ, Li X, Zeadally S. Enabling next-generation rfid applications: solutions and challenges. IEEE Comput 2008;41(September (9)):21–8.
Stephan P, Meixner G, Koessling H, Floerchinger F, Ollinger L. Product-mediated communication through digital object memories in heterogeneous value chains. In: Proceedings of the eighth annual IEEE International conference on pervasive computing and communications (PerCom). Mannheim, Germany: IEEE 2010. p. 199–207.
Stonebraker M, Abadi DJ, Batkin A, Chen X, Cherniack M, Ferreira M, et al. C-Store: a column-oriented DBMS. In: Proceedings of the 31st international conference on very large data bases (VLDB). Trondheim, Norway: ACM; 2005. p. 553–64.
Stonebraker M, Madden S, Abadi DJ, Harizopoulos S, Hachem N, Helland P. The end of an architectural era (it's time for a complete rewrite). In: Proceedings of the 33rd international conference on very large data bases (VLDB). Austria: ACM, University of Vienna; 2007. p. 1150–60.
Subramaniam S, Gunopulos D. A survey of stream processing problems and techniques in sensor networks. In: Data streams—models and algorithms; 2007. p. 333–52.
Tan CC, Sheng B, Wang H, Li Q. Microsearch: a search engine for embedded devices used in pervasive computing. ACM Trans Embed Comput Syst 2010;9(4):29.
Teymourian K, Rohde M, Paschke A. Fusion of background knowledge and streams of events. In: DEBS; 2012. p. 302–13.
Tran TTL, Sutton CA, Cocci R, Nie Y, Diao Y, Shenoy PJ. Probabilistic inference over RFID streams in mobile environments. In: Proceedings of the 25th international conference on data engineering (ICDE); 2009. p. 1096–107.
Tsatsanifos G, Sacharidis D, Sellis TK. On enhancing scalability for distributed RDF/S stores. In: Proceedings of the 14th international conference on extending database technology (EDBT). Uppsala, Sweden: ACM; 2011. p. 141–52.
Tsiftes N, Dunkels A. A database in every sensor. In: Proceedings of the ninth international conference on embedded networked sensor systems (SenSys). Seattle, WA, USA: ACM; 2011. p. 316–32.
Wang F, Liu J. Networked wireless sensor data collection: issues, challenges, and approaches. IEEE Commun Surv Tutor 2011;13(4):673–87.
Wang H, Tan CC, Li Q. Snoogle: a search engine for pervasive environments. IEEE Trans Parallel Distrib Syst 2010;21(8):1188–202.
Weikum G. Database researchers: plumbers or thinkers?. In: Proceedings of the 14th international conference on extending database technology (EDBT). Uppsala, Sweden: ACM; 2011. p. 9–10.
Welbourne E, Battle L, Cole G, Gould K, Rector K, Raymer S, et al. Building the internet of things using RFID: the RFID ecosystem experience. IEEE Internet Comput 2009;13(3):48–55.
Wu E, Diao Y, Rizvi S. High-performance complex event processing over streams. In: Proceedings of the ACM SIGMOD international conference on management of data (SIGMOD'06); 2006. p. 407–18.
Wu Y, Sheng QZ, Shen H, Zeadally S. Modeling object flows from distributed and federated rfid data streams for efficient tracking and tracing. IEEE Trans Parallel Distrib Syst 2013;24(10):2036–45.
Yan Z, Zhang P, Vasilakos AV. Research on social relations cognitive model of mobile nodes in internet of things. J. Netw Comput Appl 2014;42:120–34.
Yang Y, Wang L, Noh DK, Le HK, Abdelzaher TF. SolarStore: enhancing data reliability in solar-powered storage-centric sensor networks. In: Proceedings of the seventh international conference on mobile systems, applications, and services (MobiSys). Kraków, Poland: ACM; 2009. p. 333–46.
Yap K-K, Srinivasan V, Motani M. MAX: wide area human-centric search of the physical world. ACM Trans Sens Netw 2008;4(4):34.
Yuan J, Zheng Y, Zhang L, Xie X, Sun G. Where to find my next passenger?. In: Proceedings of the 13th international conference on ubiquitous computing (Ubicomp). Beijing, China: ACM; 2011. p. 109–18.
Zeng D, Guo S, Cheng Z. The web of things: a survey (invited paper). J Commun 2011;6(6):424–38.

Zhang Y, Pham M-D, Corcho Ó, Calbimonte J-P. SRBench: a streaming RDF/SPARQL benchmark. In: Proceedings of the 11th international semantic web conference (ISWC); 2012. p. 641–57.

Zhou Q, Simmhan Y, Prasanna VK. Towards an inexact semantic complex event processing framework. In: Proceedings of the fifth acm international conference on distributed event-based systems (DEBS); 2011. p. 401–2.

Zimmerman T, Robson C. Monitoring residential noise for prospective home owners and renters. In: Proceedings of the ninth international conference on pervasive computing (Pervasive). San Francisco, CA, USA: Springer; 2011. p. 34–49.