Title	Enabling smart societies using social internet of things: A semantic framework for real-time virtual object management
Author(s)	Shamszaman, Zia Ush
Publication Date	2019-02-04
Publisher	NUI Galway
Item record	http://hdl.handle.net/10379/15462

Enabling Smart Societies Using Social Internet of Things

A Semantic Framework for Real-time Virtual Object Management

Author: Zia Ush Shamszaman

Supervisor: Dr. Muhammad Intizar Ali

Co-Supervisor: Prof. Mathieu d'Aquin

Insight Centre for Data Analytics, College of Engineering and Informatics
Natioanl University of Ireland Galway

This dissertation is submitted for the degree of
Doctor of Philosophy

Insight Centre for Data Analytics August 2019

I would like to dedicate this thesis to my caring father Dr. A. K. M Khalequzzaman, my loving mother Rahima Pervin, my beautiful daughter Inara Shams Swaccho and my supportive gorgeous wife Safina Showkat Ara.

Declaration

I, Zia Ush Shamszaman, declare that this thesis titled, 'Enabling Smart Societies Using Social Internet of Things: A Semantic Framework for Real-time Virtual Object Management' and the work presented in this thesis are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at the Insight Centre for Data Analytics, National University of Ireland Galway.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work. I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Author: Zia Ush Shamszaman
August 2019

Acknowledgements

My PhD thesis is an outcome of a long and rugged academic journey at the Insight Centre for Data Analytics, National University of Ireland Galway. At this moment, I have arrived at a remarkable milestone that one can not reach alone without support. Yes, indeed the list of names is long that I must beholden to thank for.

First and foremost, I would like to praise and thank Almighty Allah, who has granted me the elegance, confidence and blessings to enact this endeavour on the eve of this PhD thesis completion. Then, I would like to thank my parents, my siblings Rohani Us Shamszaman and Zannatus Zohra Zerry, my daughter Inara Shams Swaccho and ofcourse my wife Safina Showkat ara for their sacrifice and support at their best capability.

My sincere gratitude and appreciation go to my thesis supervisor Dr Muhammad Intizar Ali for the patience, guideline he showed in helping me organise my research and this dissertation. I would like to thank my Co-supervisor Prof. Mathieu d'Aquin for being there with his support and suggestions when I needed. I even can not count now how many times we sit together and discuss the issues of my thesis and research and some of them repeatedly, however, they never showed me any anger or impatience that could be a normal reaction. I would also like to show my gratitude to Dr Alessandra Mileo for selecting me as a PhD student in her unit and awarding the fund for my research.

The members of my Graduate Research Committee (GRC), Dr John Breslin (Internal examiner), Dr Matthias Nickles and Dr Brian Davis have been keeping an eye on me and evaluating my work once in a year to make sure that I am on track during this period. I have received many insightful feedbacks and guidelines for my research from the committee to complete this thesis. It has been a great experience studying at Insight, NUI Galway and I appreciate all the support and discussion and fun from my colleagues and friends especially Dr Feng Gao, Thu Le Pham, Mahedi Kaysar, Andreas Kamilaris, Naomi Ono, Mohan Timilsina, Pablo Torres, Chan le Van, Duy Nguyen Duc and Brendan Smith.

Last but not least, I would like to extend my deepest and sincerest gratitude to Prof. Md Abdur Razzaque, Dr Md. Abdul Hamid and Mr. Subrato Podder for inspiring me, believing in me and bringing me in the academic research.

Abstract

A wide variety of unique smart services and applications are evolved based on Internet of Things (IoT) for resolving numerous issues in daily life, social community, agriculture, health sector, entertainment sector, city administration, environment, weather, road traffic, and many more. As a result, sensors, devices, and humans are required to associate with each other to form a new service. To get the best out of the IoT objects a platform is required to integrate different type of objects so that objects can associate with each other and can make relation like the human society.

In recent years, humans social interaction got a new edge through a profile-based online social network. The admiration of social networks and the advent of the IoT direct to a new research paradigm called Social IoT (SIoT), where real-world objects can participate in an online social network like the human social network. This effort leads to an immense possibility of unique applications for a smart cognitive society. However, it is still a challenge to explore these applications due to a lack of an adequate SIoT framework, where SIoT nodes can be managed and monitored in real-time under a cognitive framework. As these IoT nodes and services are going to co-exist with us (human), we foresee establishing of cognitive contributory skills where IoT nodes, services and even human skills can collectively form a cognitive society to share resources, information and skills. Hence, in this thesis, we propose a framework to manage and monitor the SIoT nodes intelligently and cognitively in real-time. In our proposed framework, we enable virtual representation of real-world objects known as Virtual Object (VO) and ensure their relationship semantically to compose new services by combining VOs and called Composite VO (CVO). Additionally, we identify special skills (e.g., human expertise and skill as an Abstract Object (AO).

However, building such intelligent societies automatically using SIoT is a big challenge, mainly due to the complexity of the systems and availability of a large number of nodes. In such scenarios, it is not trivial to find a suitable SIoT service node correctly to avail a service in real-time. We also propose virtual object management and selection process for the SIoT platform and QoS aware object selection using Integer Programming (IP) and Multiple Criteria Decision Making (MCDM) to find a right service at the right time.

Table of contents

List of figures

Acronyms

AMQP Advanced Message Queuing Protocol. 1

AO Abstract Object. 1

CoAP Constrained Application Protocol. 1

CPS Cyber Physical Systems. 1

CVO Composite Virtual Object. 1

DDS Data Distribution Service. 1

IERC IoT European Research Cluster. 1

IoT Internet of Things. 1

IP Integer Programming. 1

ITU The International Telecommunication Union. 1

MCDM Multiple Criteria Decision Making. 1

mDNS multicast DNS. 1

MQTT Message Queue Telemetry Transfer. 1

PO Physical Object. 1

QoE Quality of Experience. 1

QoS Quality of Service. 1

RDF Resource Description Framework. 1

REST Representational State Transfer. 1

RFID Radio-frequency Identification. 1

RSN RFID Sensor System. 1

RSP RDF Stream Processing. 1

SIoT Social Internet of Things. 1

SIoT-MF Socail Internet of Things Management Framework. 1

SNS Social Network Site. 1

SoA Service Oriented Architecture. 1

VO Virtual Object. 1

WSN Wireless Sensor Networks. 1

XMPP Extensible Messaging and Presence Protocol. 1

List of tables

Chapter 1

Introduction

The ubiquitous Internet of Things (IoT) paradigm redefines the value of data and communication by generating a huge amount of heterogeneous data over various sources. Generally, the sources of data are sensing devices and IoT devices, also known as IoT objects e.g., sensors, actuators, smartphones, smart tabs and more. IoT objects are sometimes referred as smart objects because of their ability to interact and cooperate to satisfy a common interest to improve human daily life as well as to enhance the modern society.

In the human society, people maintain a social relationship, normally, based on physical location. However, the evolve of Social Network Services(SNS) brought this relationship to the virtual world that has enabled humans to create and maintain theoretically an unlimited social community regardless the physical location. SNS consist of virtually connected people and the edges among people represent their relationship. Eventually, SNS has become one of the most popular platforms for socialization at current time.

The advent of the Internet of Things (IoT) and the admiration of Social Network Services (SNS) direct to another research paradigm known as Social Internet of Things (SIoT). It is an interesting collaboration of IoT and SNS when objects get smart, cognitive and reactive, the IoT gets social by enabling human-to-object, object-to-object and human-to-human interactions. SIoT act as a common platform for the data coming from IoT devices and SNS to allow all possible interactions among human and objects.

Handling huge amount of connected physical objects in IoT is always challenging in terms of controlling, managing and maintaining because of their heterogeneity. Consequently, physical objects require virtual representation in the cyber world called Virtual Objects (VO) [170, 108, 185] to merge with each other for composing new service based on the physical objects. Virtualizing real-world physical objects in IoT enhances flexibility to facilitate various requirements for ensuring proper application provision. Virtual Objects (VO) approach is suitable to handle these challenges [11], [18], [19]. VO is a virtual representation of real-world physical objects including IoT objects. VO concept introduces service composition at object level which is not practical when dealing with physical object. This capability refers to service mashups, service reusability, complex service creation and more.

Object virtualization bridges the gap between real-world and virtual- world. Composite Virtual Objects (CVO) are the next level VOs which are derived from the VOs by aggregating them to enhance the resilience capability and reusability to satisfy application requirements. A CVO is an intelligent mashup of semantically connected VOs and their functionalities that compose services according to requirements. Hence, CVO ensures the reusability of VOs regardless their initial contexts. In order to make the SIoT framework more wide, dynamic and intelligent, we introduce humans expertise and skills as an Abstract Object (AO) along with the real-world Physical Objects (PO). In the proposed framework we consider four types of objects,

- Real-world Physical Objects(PO)
- Virtual representation of PO called Virtual Object(VO)
- Human skill and expertise as Abstract Object(AO)
- Combination of VO and PO are called Composite Virtual Objects (CVO)

All type of objects should be connected in a way that they can collaborate with each other to create a harmonized service. The orchestration of PO and AO direct to an immense opportunity for new applications in a smart society.

SIoT leads to an immense possibility of unique applications and services for a smart cognitive society. However, it is still a challenge to explore these applications and services due to a lack of an adequate SIoT framework where SIoT nodes can be controlled, managed and monitored in near real-time under a cognitive framework. Moreover, it is necessary to develop user-friendly and intuitive tools that will motivate people without technical skills to use application and services through a context-aware and proactive framework that enable users to create their own services, it will be the users themselves who will bring out with new applications and use-cases.

1.1 Motivation

The main motivation of this research is to enable contributory societies among human and smart objects where society members can help each other by sharing their physical resources, capabilities, expertise and knowledge within the societies to execute a task. The resulting societies may introduce less expensive or free of cost voluntary solutions. We identify a few scenarios here e.g.,

- Alice has an IoT enabled smart kitchen that includes fridge and cabinet. The fridge and cabinet have the ability to order product directly from the online store and send a reminder to Alice. Hence, the review and recommendation from socially connected friends are considered to select a suitable product.
- Charlie having a problem with his car, which seems to be difficult to fix by himself. He does not know who which garage is best for this kind of solution and even it is possible he wants to avoid going to a garage because of not having his car for few days and also the expenses. His car has several sensors installed inside and based o the information collected from those sensors a report is available in the social profile. This report is circulated in the social network to find a car which had similar problems and has already been fixed. Charlie's car may get several possibilities, i.e. it may get a suggestion for a suitable garage, it may get voluntary or less expensive service from another member in the society.
- Alpha is a delivery man in a busy city who needs to cover the entire city to delivery goods frequently. It is important for him to choose a suitable route while he is driving. Hence, his car is able to collect traffic information along with a suggestion for the best route (shortest time) by exploiting the social network in advance.
- Tango's needs help for preparing his garden and paint his house this summer, additionally, he also needs a van to dump waste-garbage after cleaning the garden. Consequently, he has two options, either go to the professionals or ask for help within the society by contacting friends, relatives or neighbours one by one. The former is an expensive solution normally and the later is limited in numbers generally.
- Gamma is a traveller and visits new places frequently. Unfortunately, Gamma does not have enough money to rent places for a long time. However, His smartphone is a member of Social Network and can search for a free "house sharing" or "sleep in a couch" option according to his GPS location. We assume here that house or rooms are also a member of the social network through the IoT enabled Smart Home.

n the same way ride sharing, tractor sharing, helping people and any other resources or knowledge sharing services can be included in the SIoT enabled society. In the above-mentioned scenarios, there might be a number of capable members available at the same time who can help for a partial service or provide the entire service. It can be a human-directed service or object-directed service or a combination of human and object-directed service.

However, the problem remains in discovering, finding and selecting the right candidate at the right time. As a result it is not trivial to select the best candidates for a solution and before that all the entities in the society should be connected in a common platform, hence, the admiration of Social Network Services (SNS) and the advent of the Internet of Things (IoT) direct to a research paradigm called Social Internet of Things (SIoT) where real-world physical objects can form their own social network like human social network.

It is still a challenge to explore those applications due to a lack of an adequate SIoT framework where SIoT nodes can be controlled, managed and monitored in real-time under a cognitive framework.

1.2 Research Questions

This thesis investigate means for enabling a cognitive-contributory society in real-time by exploring socially connected objects and human expertise within a society by representing the physical world to the virtual world. To achieve this goal different activities, methods and technologies are analysed related to IoT and SIoT, i.e., object virtualization, stream processing, constraint aware selection and QoS aware selection.

Hence, the broad research question in this thesis is

"How to enable a cognitive-contributory society in real-time by exploring socially connected objects and human by virtualizing the physical objects to the virtual world?"

To answer this question few more questions need to be answered first,

- **RQ1: How to enable a better representation of the physical object to the virtual world.** *RQ1* aims at addressing the virtualization requirements of SIoT entities. It can be divided into three sub-questions:

    i) How to create a profile of objects so that machine and human both can understand?

ii) How to semantically annotate the virtual objects?

iii) How to define formal semantics of relation among virtual objects?

- **RQ2: How to select best services and objects on multiple constraints and preference in social network?**

  RQ2 aims at addressing the selection of the best candidate object and service. It can be devided into

  i) How to score different criteria of an object?

  ii) How to scale the score of different criteria of an object?

  iii) How to include user constraint and preferences?

  iv) How to improve the selection process?

- **RQ3: How to monitor and track current status of objects and update their parameter?**

  RQ3 aims at addressing the monitoring process and methods for objects.

  i) How a continuous flow of data can be monitored continuously?

  ii) How many monitoring systems are suitable for this purpose?

  iii) How to make the monitoring system adapt to satisfy dynamic requirements of heterogeneous objects?

- **RQ4: How SIoT based social recommendation can be helpful in humans life?**

  RQ4: aims at showing the social recommendation based shopping using IoT and social network.

  i) How social recommendation can help people in their shopping?

  ii) How to optimize the choices among stores, Items and recommender's?

1.3 Methodology

The research methods in this thesis consist of the following steps

1. Survey the state-of-the-art to get a view of the landscape of the relevant research areas, and understand the achievements and limitations of the current IoT, Social Network and Real-time Processing.

2. Analyse the requirements to identify concrete research problems that need to be dealt with in order to promote the cognitivity and effectiveness of SIoT.

3. A theoretical design phase to include the overall design of the SIoT platform, adaptive systems as well as the development of formalisms and algorithms and mathematical calculation used in this research.

4. Initially a constraint-based approaches are considered to select suitable VO, CVO and RSP engines.

5. Improvements of selection problem has been done to address the overlapping problems among selected objects.

6. A use-case has been discussed to show the necessity of the proposed SIoT framework.

7. Finally, the feasibility and performance evaluation phase evaluate the proposed SIoT platform with regard to applicability and domain-specific performance measures, using simulated or real-world datasets.

1.4 Overview of the Proposed Approach

The goal of this research is to enable cognitive-contributory societies by proposing a real-time virtual object management framework in SIoT i.e providing the answer of research question in section 1.2.

A real-time SIoT framework for socially connected human and objects A SIoT framework has been proposed in this research. The proposed framework offers several functionalities i.e object virtualization, real-time stream processing, semantic annotation of objects and selection of suitable objects. The entire framework enables cognitive-contributory societies where object and human can participate in a task combinedly.

Semantically annotated object virtualization Human expertise are considered as Abstract Object (AO) and the virtual counterpart of physical object is considered as (VO). The relation and connection among VO and AO are represented in an ontology model. A human profile may have more than one AO and the profile owner can set a price for each AO separately. AO and VO can be selected to execute a task according to user requirements.

A service may contain several AO and VO. VO ontology in chapter 3 provides a clear picture about their connectivity, fig... and fig shows the overview and architecture of the proposed framework.

Suitable object selection mechanism It is not trivial to select a right object when more than one capable objects are available. Situation may become even worse when the number of objects are huge; which is not unusual in IoT and Social networks. We address that problem using two different well known methods i.e, Multiple Criteria Decision Making (MCDM) and Integer Programming (IP). The candidate objects selected based quality criteria, user requirements and given score on each criteria.

Smart shopping based on social recommendations A smart shopping use-case has been described using the SIoT framework. In this work people rely on their social network friend's recommendation to get rid of fake reviews of a product. IoT enabled smart fridge and cabinet are capable to buy products online or remind the user at the right time to purchase a selected product from a selected shop.

1.5 Contributions

According to the research methodology, the research presented in this thesis lead to the following contributions in Enabling social relation among human and objects, Ontology based Virtual Object Modelling, Switching and adaptivity, and user driven service selection in order to enable cognitive-contributory societies in real-time.

Enabling social relation among human and objects. A real-time Social Internet of Things(SIoT) framework has been proposed to enable cognitive-contributory societies among socially connected human and objects. Human expertise are introduced as Abstract Object(AO) and virtual counterpart of physical objects are identified as (VO). Stream processing has also been enabled to monitor VO in real-time. The proposed framework leverage the idea of SIoT where IoT and Socail Network are integrated to bridge the gap between these two paradigm. The virtualization addresses *RQ1(i)*. A sample VO profile and its policy has been provided as an extension of [ref:WoO]. Consequently an AO profile has also been provided as an SIoT object for the first time, to the best of our knowledge.

Ontology based VO Modelling. VO and AO are annotated semantically. Consequently SIoT ontology and QoS ontology have been proposed in order to increase expressivity and re-usability. A registry logs the VO and AO profile following the ontology model, moreover it also preserves the newly formed CVO (Composite VO: combination of more than one VO and/or AO). Semantic ontology based modelling addresses *RQ1(ii),(iii)*.

Selection Mechanism. Two mechanisms are adopted to address user constraint based suitable object selection among available capable objects. They are MCDM and IP solution. Selection problem solution address *RQ2(i),(ii),(iii) and (iv)*.

Real-time Monitoring. An adaptive real-time monitoring approach for SIoT objects has been proposed based on RDF stream processing (RSP) engines. To make the RSP suitable for dynamic objects and its requirements, an adaptive layer has been introduced on top of available engines. The adaptive layer selects the suitable engines by following MCDM. Real-time monitoring addresses *RQ3(i),(ii) and(iii)*.

Utilization of Socially connected objects. In addition to the above contributions, the usage of the SIoT framework in smart shopping is presented. A practical problem of avoiding fake review is addressed along with the recommendation and reminding service from SIoT framework based applications. The use-case addresses *RQ4(i),(ii)*.

1.6 Thesis Outline

The research in this thesis is introduced in this section.

- *Chapter 2* discusses the preliminaries and background technologies of this research. This chapter is divided into IoT, Middleware, QoS based Web service composition and object virtualization sections.
- *Chapter 3* discusses the state-of-the-art technologies that have been considered in this thesis i.e Semantic Web, Semantic stream data processing, Social IoT.
- *Chapter 4* introduces a real-time SIoT framework, its architecture and also presents the object virtualization and the relation and ontology for virtual objects. This chapter includes the detail description of each component of the proposed framework.

Additionally, this chapter asl o showcases a use-case of the SIoT framework. A smart shopping scenario has been explained where people can get rid of fake reviews and a few more smart features like recommendations and reminders.

- *Chapter 5* showcases the approaches to address a suitable object selection problem. MCDM and IP have been used to select a suitable object where more than one capable objects are available. In these approaches, user requirements are considered as the constraints and the available capable objects are considered as candidate objects. QoS is also explained in this chapter.
- *Chapter 6* presents the real-time stream processing capability in the proposed framework. This chapter also addresses the suitable stream processing engine selection problem. MCDM has been considered again to select a suitable engine.
- *Chapter 7* concludes the achievements and limitations of this research. and the most importantly the open research question for future work.

1.7 List of Relevant Publications

1. Shamszaman, Z. U., & Ali, M. I. (2019). Enabling cognitive contributory societies using SIoT: QoS aware real-time virtual object management. Journal of Parallel and Distributed Computing, 123, 61-68.
2. Shamszaman, Z. U., & Ali, M. I. (2018). Toward a Smart Society Through Semantic Virtual-Object Enabled Real-Time Management Framework in the Social Internet of Things. IEEE Internet of Things Journal, 5(4), 2572-2579.
3. Shamszaman, Z. U., & Ali, M. I. (2017, September). On the Need for Applications Aware Adaptive Middleware in Real-Time RDF Data Analysis (Short Paper). In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems" (pp. 189-197). Springer, Cham.
4. Ali, M. I., Ono, N., Kaysar, M., Shamszaman, Z. U., Pham, T. L., Gao, F., & Mileo, A. (2017). Real-time data analytics and event detection for IoT-enabled communication systems. Web Semantics: Science, Services and Agents on the World Wide Web, 42, 19-37.
5. Shamszaman, Zia Ush, Muhammad Intizar Ali, and Alessandra Mileo. (2015) "On the need for adaptivity in RDF Stream Processing.", RSP Workshop in ESWC.

6. Shamszaman, Z. U. (2014, September). Adaptive Stream Query Processing Approach for Linked Stream Data:(Extended Abstract). In Web Reasoning and Rule Systems: 8th International Conference, RR 2014, Athens, Greece, September 15-17, 2014. Proceedings (Vol. 8741, p. 251). Springer.

1.8 Summary

In this chapter, we have presented a few motivating scenarios of this thesis. The higher level and lower level research questions are also explained in this chapter which is going to be addressed in this thesis. We have also described the methodology we have followed to conduct this research. The main contribution of this thesis and the overview of the proposed approach are presented. This entire chapter provides a glimpse of this research.

Chapter 2

Background

In this chapter, the basic concepts and methodology related to this thesis are presented. In particular, it describes the relevant technologies, methods and their current status.

2.1 Internet of Things (IoT)

The Internet of Things (IoT) envisions the concept of pervasive and ubiquitous computing including sensors, actuators, mobile devices and even product information tags. The scope of IoT covers, all these things also known as smart things/objects are uniquely addressable. These smart objects are also capable to interact with their environment, and react to an event with other things/objects to accomplish the assigned tasks [136, 177, 16, 164, 170]. Internet of Things (IoT) research is still not in a matured stage and yet to provide a single formal definition. Generally, IoT can be realized through three different perspectives 1) Internet-oriented; 2) things-oriented (smart things/objects or sensors) 3) semantic oriented (knowledge) [16].

The very first definition of the IoT was derived from RFID tag based settings where RFID tags were things [16]. Based on the RFID community describe IoT as "The worldwide network of interconnected objects uniquely addressable based on standard communication protocols" [67] . Figure 2.1 shows the European research cluster of IoT (IERC) definition, where IoT is connecting anything to anyone at any time from any place using any network and any service [175, 183, 136, 45, 197, 75]. A similar type definition is provided by "The International Telecommunication Union (ITU) “From anytime, anyplace connectivity for anyone, we will now have connectivity for anything” [149]. Semanti-

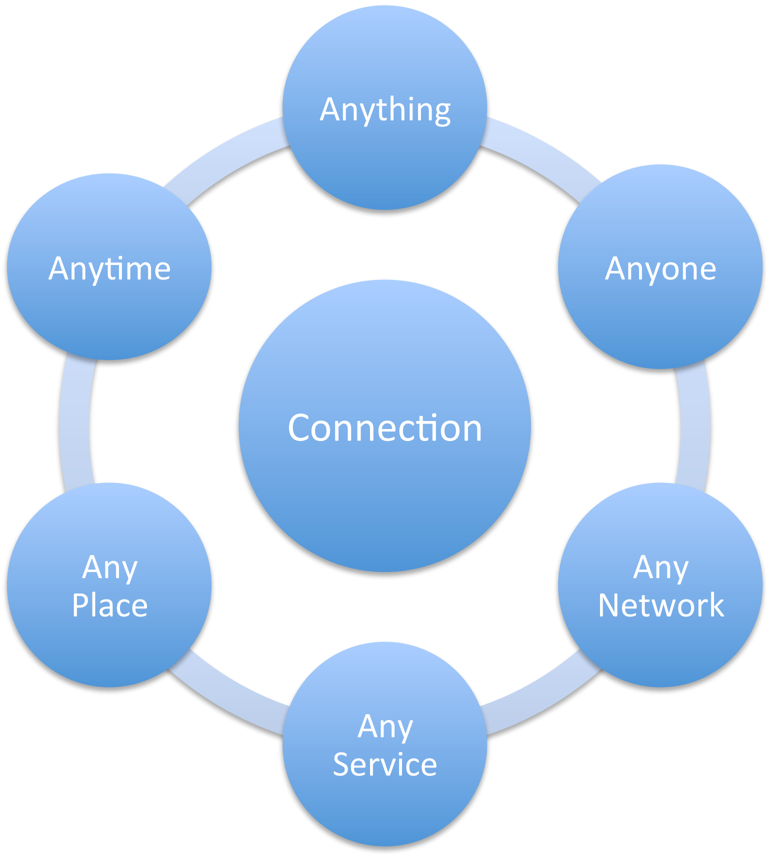

Fig. 2.1 Main objective of IoT

cally, IoT implies uniquely addressable interconnected objects across the globe, based on regular communcation protocols [67].

Many definitions of IoT do not explicitly spotlight the industrial perspective of IoT (IIoT). However, world top businesses are giving exclusive focus as well as producing considerable investments to include the IoT to their manufacturing solutions. Despite the fact that they normally use different phrases like "Smarter Planet" by IBM, "Internet of Everything" by Cisco and "Industrial Internet" by GE digital, their primary goal is using IoT to improve manufacturing by cutting down uncertain machine downtime as well as considerably minimizing electricity bills plus a selection of additional likely advantages [213, 2, 91, 5, 4]. The IIoT means industrial things, or objects, equipped with sensors, automatically communicating through a network, without any human interaction, to exchange information and capture the insight from available data with the assistance of smart analytics [5]. The meaning of "things" or "object" in the IoT concept is extremely vast and has a multitude of actual physical components. These include objects like personal items we hold, e.g., smartphones, digital cameras, and tablets.

The IoT includes billions of objects that are capable to collect heterogeneous data which were not easily accessible earlier such as Wireless Sensor Networks (WSNs) and Machine-to-Machine (M2M) can only extract local data containing particular information from the objects [215]. However, IoT can provide widespread, and historical information by collaborating with different Intranet of Things regardless of their heterogeneity of communication technologies and deployment goals.

Advancements in the Internet of Things (IoT) accompany a smart life where real-world objects, including sensing devices, are interconnected with each other to create services for human society. The Web representation of smart objects empowers innovative applications and services for various domains. Eventually, a wide range of IoT based services and applications are evolved to solve humans daily life problems at the individual level and also at the organizational level.

Many industrial, standardization and research bodies are presently active in the exercise of advancement of IoT to fulfil the highlighted technical needs. This section provides a summary of the present status of the art of the IoT paradigm. Additionally, a handful survey works on IoT research is also available Li et al. [122], Whitmore et al. [190], Perera et al. [151], Al-Fuqaha et al. [8], Da Xu et al. [58], Atzori et al. [17], Lin et al. [128].

Several research communities are involved in building IoT systems prototype [193, 59]. Nevertheless, a majority of the systems focus on the specific applications and applied within intranet or extranet, and also have no interaction with each another. Depending on the functions and features of IoT we can see the interconnection is a crucial architecture issue. Unfortunately, these IoT systems are not "IoT," however the "Net of Things," or can be counted as "Net of Devices," and even the interactions among these extranets and intranets are missing [178, 193]. IoT must consider all the things/objects in the large scale, where different networks must coexist, and therefore are competent to have interaction with each another via several gateways and middlewares, backed by the complicated control plane [138]. One particular perspective is the fact that a common network infrastructure which combines different networks must be created, and most IoT based systems or maybe applications are able to provide the services of theirs by effectively sharing community resources and info across the generalized community infrastructure. For instance, in smart cities, [34, 203], a common network infrastructure can be implemented and covers all areas in a city. Applications such as smart grid, smart healthcare, smart transportation, etc. can easily share their personal network infrastructures to allow data collection and also information delivery. In this particular vision, every aspect that's interconnected in the system is able to be recognized because most applications can easily communicate with one another readily and also discuss the resources successfully. The implementation of the common network infrastructure is able to minimize the price of network deployment also [203].

To get a common network infrastructure, the development of IoT with respect to architectures, enabling technologies, as well as potential difficulties. Recently, a number of published survey assessed the IoT technologies from numerous aspects. For instance, the survey work carried out by Atzori et al. [17] given the enabling communication solutions as well as various visions of IoT, that helps to address this particular area have a main understanding of IoT. The survey work carried out by Al Fuqaha et al. [190] presented potential applications, protocols, and enabling technologies of IoT, in which the horizontal introduction of IoT was depicted as well as the major IoT problems have been presented to mention the succeeding directions. Right now there are also a number of investigation work dedicated to secrecy and privacy issues in IoT. For example, the survey work carried out by Andrea et al. [13] given the security vulnerabilities and issues in IoT from the perspective of physical systems, networks, and applications, and also thought the security and secrecy issues in technologies related to actual physical methods, software, networking, and encryption. The survey work carried out by Sha et al. [188] presented difficulties problems as well as possibilities in IoT. Along with the above mentioned survey papers, Botta et al. [40] thought the integration of IoT as well as cloud computing. Additionally, Wu and as Zhao [193] proposed a novel IoT infrastructure, that could be designed and as well as recognized by existing Internet technologies, and also meets different needs of IoT. Even though a numerous works are performed, many existing surveys have just highlighted on certain aspects of IoT.

To bridge the gap, Jie L et al [129] reviewed the existing initiatives on IoT then provide the integration of fog/edge computing as well as associated problems in IoT. Particularly, they have performed an extensive overview of IoT with regard to architectures, enabling technologies, privacy and security issues, and also provide the basis of fog/edge computing based IoT as well as their applications. Especially, the relation between Cyber Physical Systems (CPSs) and IoT is investigated. It is Worth mentioning here that each IoT and CPS highlight the interactions between the cyber world as well as the actual physical world, and therefore are readily confused with each other. Additionally, the big difference between IoT and CPS hasn't been clearly distinguished before. The comprehensive relation between IoT and CPS is able to assist newcomers to comprehend the idea and also options that come with IoT. They have presented IoT as a multilayer architectures, split into the perception layer, networking layer, service layer and application layer. According to the multilayer architecture, enabling technologies as well as opened issues in each level are presented [128].

2.1.1 Internet of Things (IoT) Architecture

We have analysed several existing IoT architectures in this section. We consider mainly two types architecture Three layer architecture and SoA-based architecture.

A. Three Layer Architecture

Generally, the structure of IoT is split into three primary layers [133], a) perception layer; b) network layer; along with c) application layer, which are discussed below in detail.

Perception Layer: It is also known as the sensor layer, is implemented as the bottom layer in IoT architecture [21]. The perception layer interacts with physical devices and components through smart devices (RFID, sensors, actuators, etc.). Its main objectives are to connect things into IoT network and to measure, collect and process the state information associated with these things via deployed smart devices, transmitting the processed information into upper layer via layer interfaces.

Network Layer: It is also known as the transmission layer, is implemented as the middle layer in IoT architecture [119]. The network layer is used to receive the processed information provided by perception layer and determine the routes to transmit the data and information to the IoT hub, devices, and applications via integrated networks. The network layer is the most important layer in IoT architecture because various devices (hub, switching, gateway, cloud computing perform, etc.), and various communication technologies (Bluetooth, Wi-Fi, long-term evolution, etc.) are integrated into this layer. The network layer should transmit data to or from different things or applications, through interfaces or gateways among heterogeneous networks, and using various communication technologies and protocols.

Application Layer: It is also known as the business layer, is implemented as the top layer in IoT architecture [8]. The application layer receives the data transmitted from the network layer and uses the data to provide the required services or operations. For instance, the application layer can provide the storage service to backup received data into a database, or provide the analysis service to evaluate the received data for predicting the future state of physical devices. A number of applications exist in this layer, each having different requirements. Examples include smart grid, smart transportation, smart cities, etc. [179, 194].

B. SoA-based Architecture

Normally, SoA is a component-based design, that may be created to link various practical devices (also referred to as services) of an application by interfaces as well as protocols [17, 137, 57]. SoA is able to concentrate on developing the workflow of coordinated services, and also enable the reuse of software as well as hardware pieces, enhancing the feasibility of SoA for use in developing IoT architecture [17, 59]. Consequently, SoA can be incorporated into IoT architecture, in what information services offered by the network layer as well as the application layer in the standard three-layer architecture could be extracted as well as create a brand new layer, specifically the service layer (also referred to as user interface layer or maybe middleware layer). Consequently, in a SoA based IoT architecture, four layers are available and then communicate with one another [176], these being the notion layer, service layer, network layer, and program layer. It's well worth noting that, in certain present attempts, the program layer is split into two sublayers, specifically service composition sublayer as well as program control sublayer. Additionally, the business layer is obtained from the application layer and also functions as the top of the layer of the application layer to offer complicated service requests.

In the four-layer SoA based IoT architecture, the perception layer is done at the bottom layer of the architecture and utilized to calculate, collect, and extract the information related to bodily gadgets [103]. The network is utilized to figure out routes and supply information transmission help via integrated heterogeneous networks [17, 97]. The assistance layer is situated in between the network as well as application layer, providing solutions to allow for the application layer [17]. The assistance layer includes service discovery, service management, service composition, and service interfaces. Below, service find is utilized to learn desired service requests, service structure is utilized to have interaction with the connected objects, and also divide or perhaps incorporate offerings to meet up with service requests in an effective manner, service control is utilized to control and find out the trust mechanisms to meet service requests, as well as service interfaces are utilized to help interactions among all provided services. The software layer is utilized to allow for the program requests by users. The software layer is able to help support a selection of uses, including smart grid, smart cities, smart transportation, etc.

2.1.2 Enabling Technologies and Challenges

Depending on the architectures stated previously, IoT may be recognized with numerous enabling solutions. In this particular area, the four layer SoA based IoT architecture is considered as a good example to provide the appropriate enabling issues as well as solutions in each level.

A. Perception Layer: In the perception layer, the primary purpose is usually to determine as well as track items. In order to do this performance, the following solutions might be applied. *RFID:* Typically, RFID, as being a non-contact communication technology, can be used to determine as well as track objects with no contact. It assistance information exchange via radio signals over a brief distance [13, 211]. The RFID based structure comprises of RFID tag, RFID viewer, and antenna [71]. RFID tag is usually a microchip placed on an antenna. Every RFID tag is connected to an item and also has the distinctive identification number of its. An RFID reader is able to determine an item and get the corresponding info by querying towards the attached RFID tag through right signals [113]. An antenna is utilized to transmit signals between RFID tag as well as RFID audience. In comparison along with other technologies, RFID has got the following advantages [52, 178] (fast scanning, low cost, small size, security, noncontact reading, large storage, reusability, durability, etc.). Due to these advantages, RFID might be helpful in the belief level of IoT to determine as well as track items as well as exchange information.

Wireless Sensor Networks: Wireless Sensor Network (WSN) is able to perform an extremely crucial part in IoT [66, 195, 142, 135, 115, 201]. WSN is able to monitor as well as monitor the condition of products, and also transmit the condition information to the management centre or maybe sink nodes via several hops [7, 125]. Consequently, WSN may be viewed as the additional bridge in between the real life as well as the cyber community [192]. In comparison along with other technologies, WSN has a variety of advantages, low cost, small size, reliability, dynamic reconfiguration, including scalability, along with decreased power consumption. These benefits help WSN to be incorporated into different aspects with several needs. Discover each WSN and RFID may be utilized for information acquisition of IoT, as well as the distinction is the fact that RFID is primarily employed for item identification, while WSN is primarily used for the perception of real world actual physical details linked to the surrounding atmosphere [178].

Others: Barcode, likewise denoted 1 D code, shops the info in many black colored lines as well as gray spacings. These lines as well as spacings have various widths, structured in a 1-D or linear path, and therefore are set up with unique encoding rules [98]. The info contained in the barcode may be checked out by a machine which goes through the barcode with an infrared beam [141]. A 2-D code captures the info through the use of white and black pixels presented on the airplane, where black colored pixel belongs to a binary of "1" and gray pixel represents a binary of "0" [98]. With unique encoding rules, the white and black pixels are able to keep a substantial level of info. In comparison and have a barcode, 2-D code has got the gain of higher info material, high robustness, high reliability, etc.[98]. Additionally, RFID sensor system (RSN) is an integration of RFID program as well as sensor system. In an RSN, sensor system is able to cooperate with RFID phone system to determine as well as monitor the condition of items [98]. In an RSN, tiny RFID based sensing equipment as well as RFID reader are implemented, the place that the RFID reader operates like a sink node to produce information and offers strength for community operations.

B. Network Layer: The network layer is utilized to determine routing, and also supply information transmission help via integrated heterogeneous networks. In the following, several protocols which can enable the secure and reliable interaction of IoT are presented.

1) IEEE 802.15.4: IEEE 802.15.4 is a protocol created just for the bodily coating and also the MAC level in wireless private area networks (WPANs) [8, 84]. The objective of IEEE 802.15.4 is focusing on low rate WPANs, if the lower price connections of all things in an individual location with lower power usage, very low speed transmission, along with affordable [54]. IEEE 802.15.4 protocol stack is grounded on open process interconnection version, in that every level just implements elements of transmission functions, minimizing levels are able to supply service to top layers. IEEE 802.15.4 is able to support bands of 868/915M as well as 2.4 GHz, as well as the information transmission price on these bands can do twenty, forty, and also 250 Kb/s, respectively, [8]. IEEE 802.15.4 is a foundation for a lot of wireless communication solutions and protocols, like ZigBee [110], WirelessHART [109], etc.

2) 6LoWPAN: Low-power WPANs (LoWPANs) are structured by a lot of inexpensive products attached by wireless communications [178]. In comparison along with other kinds of networks, LoWPAN has a selection of benefits (small package sizes, energy that is low, low bandwidth, etc.) [178]. As an enhancement, 6LoWPAN protocol was created by com bining LoWPAN. and IPv6 In 6LoWPAN, IPv6 packets will be transmitted over IEEE 802.15.4 networks [147]. Due to the low price as well as decreased power usage, 6LoWPAN is ideal to IoT, where a lot of affordable products are included. 6LoWPAN have a number of benefits, including an excellent compatibility and connectivity with history architectures, ad-hoc self-organization, low-energy consumption, etc.

3) ZigBee: ZigBee is a wireless network engineering, created for short-term interaction with low energy usage [148]. Inside ZigBee protocol, five layers are included: the actual physical level, the MAC level, the transmission layer, the system level, so the application level [178]. The benefits of ZigBee networks include minimal power usage, price that is low, low data fee, reliability, low complexity, and protection. ZigBee networking is able to help support several topologies, tree, including star, as well as mesh topologies [32].

4) Z-Wave: Z-wave is a short-term wireless communication technology with all the benefits of the cost that is low, very low power usage, along with good reliability [148]. The primary goal of Z wave is providing dependable transmission between a management system plus one or maybe more end devices, along with Z wave is ideal for the system with low bandwidth. Discover that at most 232 nodes (slaves) may be incorporated in a Z wave system, plus most nodes (slaves) will be managed by the controller and also have routing ability [148, 178]. Z-wave system supports the powerful routing technology, so every servant shops a path list in the mind of its, that is current by the controller [89]. Even though each of Z-wave and ZigBee supports the short-range wireless communication with the cost that is low as well as decreased power usage, you will find a number of variations between them. The primary distinction between Z-wave and ZigBee is the frequency band operated in by the bodily level. Inside ZigBee, the frequency band of the bodily coating is generally 2.4 GHz, even though the frequency band of Z wave is under one GHz (908.42 - 868.42 MHz) [178]. The ZigBee system is able to help support end products (slaves) as many as 65000, even though the Z wave system is only able to support 232 end devices (slaves) [178]. In comparison with ZigBee architecture, Z wave is easy in setup.

5) Message Queue Telemetry Transport: Utilizing the publish/subscribe method, message queue telemetry transport (MQTT) is a messaging process, that is utilized to obtain calculated details on remote sensors and transmit the information to servers [8]. MQTT is a lightweight and simple process and also supports the system with substantial latency and low bandwidth. MQTT is usually applied in numerous platforms to link conditions in IoT into the Internet, and hence MQTT may be utilized as a messaging protocol involving servers and sensors/actuators, making MQTT play a crucial part in IoT.

6) Constrained Application Protocol: Constrained software protocol (CoAP) is a messaging protocol dependent on representational state transfer (REST) structure [8, 39, 85]. Simply because the majority of the equipment within IoT are methods constrained (i.e., little storage as well as minimal computing capability), HTTP can stop being utilized in

IoT, thanks to the complexity of its. In order to conquer the problem, CoAP was suggested modifying several HTTP features to meet up with the demands for IoT. Typically speaking, CoAP will be the application level protocol in the 6LoWPAN process stack, and also seeks to allow methods constrained products to attain RESTful interactions. The team interaction as well as push notification are dependent on CoAP, but broadcasting isn't. Source observation, block-wise resource transport, source discovery, interaction with HTTP, and safety measures are the key features offered by CoAP [8, 85].

7) Extensible Messaging and Presence Protocol: Extensible messaging, as well as presence protocol (XMPP), is a quick messaging protocol dependent on XML streaming protocols [8, 163]. XMPP inherits capabilities of XML protocol, to ensure that XMPP has amazing scalability, dealing with, as well protection abilities, and may be utilized for multiparty chatting, speech as well as video streaming, and telepresence. In XMPP, the following three roles are included: a) client; b) server; along with c) gateway and bidirectional communication, is backed between two people of these three functions. Especially, the server is able to accomplish the performance of email and link control routing, the gateway is utilized to allow for the steady interaction among heterogeneous methods, as well as the customer could be hooked up to the server according to TCP/IP process and also transmit context dependent on XML streaming protocol. Consequently, XMPP may be utilized in IoT to allow for the item to object interaction with XML based text messages.

8) Data Distribution Service: Information division service (DDS) is a publish/subscribe protocol for supporting high-performance device-to-device correspondence [8, 199]. DDS was created by object-manage-group [199] and it is a data-centric process, where multicasting may be supported to attain top reliability and excellent QoS. The broker less publish/subscribe architecture makes DDS ideal for real-time constrained IoT as well as device-to-device communications [8]. Additionally, DDS is able to achieve a better scalability.

9) Advanced Message Queuing Protocol: Sophisticated email queuing protocol (AMQP) is an open standard email queuing protocol utilized to give message service (queuing, routing, safety measures, reliability, etc.) in the application level [8, 86]. AMQP concentrates on the message oriented locations and also could be looked at as a message oriented middleware protocol. Utilizing AMQP, customers are able to attain healthy reception with note middlewares, even when these customers, as well as middlewares, are made by various programming languages. Additionally, AMQP tools different kinds of information exchange architectures, which includes retailer and advanced, post and also subscribe context-based routing, message queuing, message distribution, and point-to-point routing [165].

10) Others: Besides the transmission protocols, communication protocols, and messaging protocols, many other protocols are able to play crucial roles in IoT also. For instance, multicast DNS (mDNS) is able to help support the title resolution in IoT apps [8, 104]. DNS service find can be utilized by customers to learn the preferred products in a unique community via mDNS [8, 161]. Routing protocol for power that is low as well as lossy networks is a link independent routing process, that could be deployed during resource constrained nodes to figure out routes over lower power and lossy backlinks [8, 200, 191]. Even though these protocols could be incorporated into IoT, enhanced protocols with increased protection, reliability, as well interoperability abilities have to advance the improvement of IoT.

C. Service Layer The service layer is placed between the network layer and the application layer and offers secure and efficient services to applications or objects. In the service level, the following enabling solutions must be incorporated to make sure that the program could be provided efficiently: user interface engineering, service control technology, middleware know-how, and sharing technology and resource management.

1) Interface: The interface should be designed in the service layer to make certain the secure and efficient info exchange for communications among applications and devices. Additionally, the interface must proficiently manage the interconnected products, including unit connection, device communication, device disconnection, and unit procedure [59].

An interface profile (IFP) is required as a service standard, that may be utilized to facilitate the interactions amongst services offered by different applications or devices. To attain an effective IFP, common plug and play ought to be implemented [83, 94, 59]. Several efforts on IoT interface have been conducted for the development of IoT. For example, SOCRADES integration architecture that can be utilized to provide meaningful interactions between applications and services [94, 158]. As the improvement of SoA IoT, service provisioning method has got the ability to enable interactions with applications and services [59, 214]. Even though a selection of IoT-user interface solutions is already available, applying more secure, scalable interface solutions with low-cost still remain a challenge for the development of IoT.

2) Service Management: Service management might effectively discover the equipment in addition to applications, and also schedule proficient and reliable services in order to meet upwards with requests. A service might be checked out as a behaviour, like the collection, exchanging, and the storage of data or perhaps a connection of these behaviours to achieve a unique goal [17, 137]. The service and requirements in IoT may portray one to may relationship at both sides, it means, a single service can satisfy multiple requirements and sometimes multiple service is required to satisfy a single requirement. The program may be split into 2 groups in IoT: a) main services as well as b) secondary service [59]. The main service, likewise called the standard service, can expose the main functionalities of applications or devices. In comparison, the secondary service is able to achieve the auxiliary functionalities depending on the main service or any other secondary service.

In order to conceal the implementation information of services and also create these services be compatible for implementation in heterogeneous devices and programs, SoA happens to be employed to incorporate services. By means of this particular, the dependability as well as consistence of solutions are usually supplied [59, 130]. For instance, OSGi platform started by a powerful SoA architecture is a good modular platform to deploy services. To deploy a SoA based service, the program composition platforms must be created first, after which the functionalities as well as correspondence features of products must be abstracted. Lastly, a common set of services must be provisioned [17, 59]. For SoA based service, each service provided by a device or software may be viewed as a regular service, which is usually easily and effectively utilized in different heterogeneous devices as well as uses with no change. This way, needs in SoA based IoT can be happy more fast and efficient [59, 128].

Additionally, raw data in IoT are gathered up by smart products (RFID, etc.), sensors, and the majority of these clever gadgets are resource constrained and can't crop electricity from the planet. Consequently, an energy saving program might be of interest in resource management [159]. We have seen a selection of projects on energy efficiency as well as energy management in sensor networks, which includes systems in order to improve the lifetime of receptors through harvesting electricity from distributed energy resources [172], schemes to lower the power of sensors through duty cycle scheme [146], energy-based routing protocols to balance the power usage and then to boost the lifespan of the sensor system [201, 191]. Even though these initiatives are able to work nicely on energy saving as well as management, a system that's ideal for IoT system infrastructures made up of heterogeneous networks is an unresolved struggle for future investigation too.

2.1.3 Cyber Physical Systems (CPS)

In this particular area, the relation and differences between IoT and CPS is clarified. In the following, we initially provide the overview of CPS then talk about the primary differences with IoT.

A. Synopsis of CPS

CPS is described as the system which could effectively incorporate physical component and cyber components with the integration of the contemporary computing as well as communication technologies [6, 192] aiming to transforming the technique of interaction of all the human, physical world and cyber world. CPS focuses on the interactions between physical components and cyber and has an objective of making the monitoring as well as controlling of physical elements security, efficiency, and intelligence by using cyber components [43].

In CPS, "cyber" means utilizing the contemporary sensing, computing, as well communication technologies to successfully monitor as well as limit the actual physical elements, while "physical" means the real-world physical objects, and also the intricacy is mirrored by "system" as well as variety. According to the clarification, a CPS comprises several heterogeneous distributed subsystems [99]. Like the development of IoT, CPS is created in several places [99, 126, 127], including smart grid, sensible transportation, etc. As revealed in [43], the CPS is the integration of physical elements, receptors, actuators, communication networks, and control facilities, where sensors are deployed to a certain and monitor the condition of actual physical elements, actuators are deployed to make certain the attractive activities on physical components, and communication networks are used to provide calculated information as well as responses commentary along with sensors, actuators, and control facilities. The management centres are utilized to assess calculated information and delivered feedback instructions to actuators, making sure the process works in the ideal states [43, 128, 198].

B. Relation and Difference between CPS and IoT

Based on the overview of CPS, we know that both CPS and IoT aim to achieve the interaction between cyber world and physical world [148]. Particularly, CPS and IoT can measure the state information of physical components via smart sensor devices without human's input. Meanwhile, in both CPS and IoT, the measured state information can be

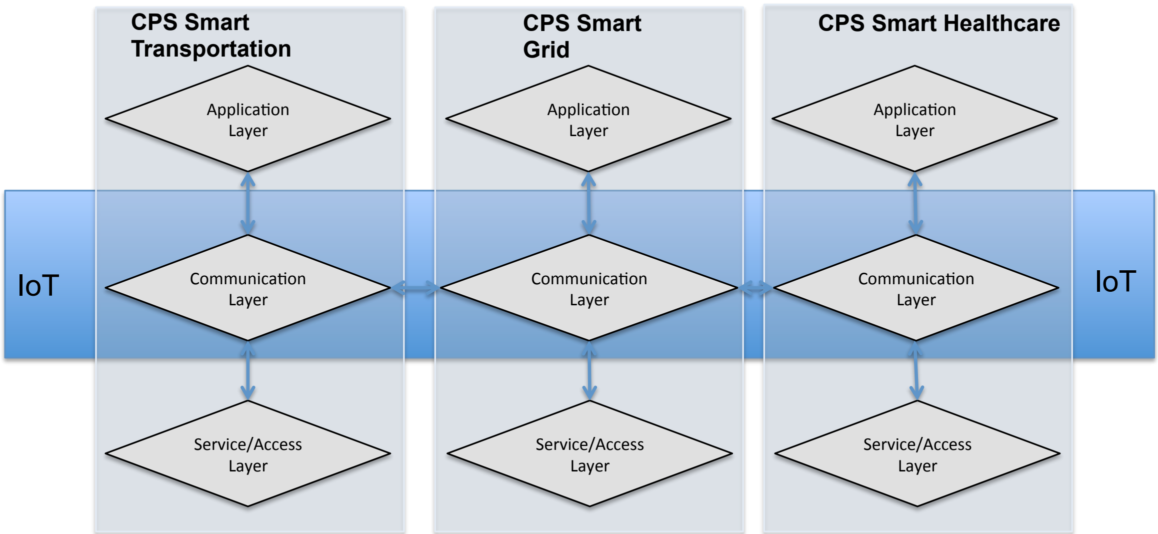

Fig. 2.2 IoT and CPS synthesis

transmitted and shared through wired or wireless communication networks. After the analysis of measured state information, both CPS and IoT can provide secure, efficient, and intelligent services to applications. The existing efforts on CPS applications and IoT applications have been expanded to similar areas (smart grid, smart transportation, smart city, etc.) [128].

Because of the similarities between CPS and IoT, it is an urgent need to clarify the difference between CPS and IoT so that newcomers may enter this complex discipline easily. Nonetheless, few existing efforts clearly identify the difference between CPS and IoT, and several efforts have even considered the CPS and IoT to be the same concept. Thus, to fulfil this gap, the difference of CPS and IoT is clarified below. [136]. As mentioned above, the essence of CPS is the system and the main objective of CPS is to measure the state information of physical devices and ensure the secure, efficient, and intelligent operation on physical devices. In CPS, the sensor/actuator layer, communication layer, and application (control) layer are present. The sensor/actuator layer is used to collect real-time data and execute commands, communication layer is used to deliver data to upper layer and commands to lower layer, and application (control) layer is used to analyze data and make decisions. Figure 2.2 illustrates the three layers in CPS. From this figure, we can see that CPS is a vertical architecture.

In contrast, IoT is a networking infrastructure to connect a massive number of devices and to monitor and control devices by using modern technologies in cyber space. Thus, the key of IoT is “interconnection.” The main objective of IoT is to interconnect various networks so that the data collection, resource sharing, analysis, and management can be carried out across heterogeneous networks. By doing so, reliable, efficient, and secure services can be provided. Thus, IoT is a horizontal architecture, which should integrate communication layers of all CPS applications to achieve interconnection, as shown in Figure 2.2. Notice that, the interconnection of various networks is not only limited to physical connections.

Control plane (interfaces, middleware, protocols, etc.) should be designed to ensure that data can be efficiently delivered across different kinds of networks and shared. For instance, in a smart city, networks of smart weather forecasting, smart transportation, and smart grid should be interconnected and interact with each other. Data from smart transportation and smart weather forecasting should be processed and extracted and used by the smart grid to determine the states and brightness of street-lamps to ensure efficient use of energy resources, as well as traffic safety at night.

In fact, control plane in IoT is much more complicated compared to that on online and has long been ignored by nearly all if not every. Lately, several initiatives are centred on the management aeroplane found IoT. For instance, Wu, as well as Zhao [193], recommended an IoT structure, specifically WInternet, that concentrates on interconnecting different Net of Things right into a large scale worldwide community. Inside WInternet, the inner structure of nodes was innovated with lodged computing ability to guarantee that vital programs are able to meet up with actual physical space. Additionally, protocols (netlet computation as well as pipe correspondence protocol) have been developed to meet up with the demands of IoT apps.

To summarize, the basic difference between CPS and IoT is that, CPS is considered as a system, while IoT is considered as "Internet." The common requirements for both CPS and IoT are real-time, reliable, and secure data transmission. The distinct requirements for CPS and IoT are listed as follows: for CPS, effective, reliable, accurate, real-time control is primary goal, while for IoT, resource sharing and management, data sharing and management, interface among different nets, massive-scale data and big data collection and storage, data mining, data aggregation and information extraction, and high quality of network quality of service (QoS) are important services.

In reality, among the most symbolic applications which integrate IoT and CPS is smart cities, in which a few CPS uses work together, including smart grid, smart healthcare, smart transportation, etc. As shown in Figure 2.2, the correspondence levels of all applications are interconnected as a single community to supply service for smart cities.

2.1.4 IoT and Fog/Edge Computing

Integration of IoT with Fog/Edge computing became a popular recent trend. In this section, we discuss analyse the present status of Fog/Edge computing integrated IoT.

A. Overview

The information produced by the points demands big data to gather as well as process the info that's made as well as gathered, and transform it into one thing that's helpful. Large details demands the assistance of IoT due to the difficulties of substantial actuating and sensing information supported by IoT (smart grid, wise transportation, etc.). Additionally, the information collected in IoT applications are unstructured data, and require further analysis to extract information that is useful. Big data and the IoT are able to work nicely with one another. A real world case is United Parcel Service (UPS), that is among the biggest delivery businesses on the planet [131]. UPS deploys sensors to gather information (which will be the IoT application) and perform the big information analysis to minimize cost and boost delivery effectiveness. The sensors are deployed on the delivery vehicles and gather the monitoring the info (mileage, fuel cost, speed, etc.).

As IoT is now the next technology revolution, it is going to affect great data in factors of information storage, information processing, and analytics. In IoT, constant streams of data will influence the information storage capacity in numerous organizations. Extra details centres will likely be required to cope with a ton of data collected from IoT apps. A solution that is possible is moving the information on the cloud by leveraging the platform as being a program. When a company selects a technology for carrying out large data processing as well as analytics, the dynamics of the IoT data has to be considered. Hive and hadoop can be used to deal with great data. Nevertheless, for information collected by IoT program, NoSQL document databases (the Apache CouchDB, etc.) might be suited [65]. This's since the NoSQL document databases are able to provide low latency and high throughput. Additionally, Apache Kafka is but one IoT tool for intermediate email brokering It may be utilized for the real time stream processing. The protection of big data will additionally be influenced by IoT [65].

B. Fog/Edge integrated IoT

Cloud computing has become a mature technology utilized to pro vide computing services or maybe information storage on the internet, and the majority of the huge IT businesses

(Amazon, Google, IBM, etc.) are hosting cloud services. Cloud computing offers the advantages of freedom, efficiency, and power to use and store information. Nevertheless, when cloud computing is utilized in IoT, brand new issues will show up. In most IoT/CPS applications, information from an enormous number of items plus objects spanning a big geographical area have to be saved, processed, and ana lyzed effectively. In order to satisfy the gap, fog/edge computing has the capacity to extend cloud computing being nearer to the points it sup ports [174]. Rather than performing all of the computation in the middle of the cloud, fog/edge computing is able to offer computing as well as storage service to products (nodes) in the edge of the system.

A fog/edge computing node is some system unit with all the ability of network connectivity, computing, and storage (routers, switches, video surveillance cameras, etc.), servers, These units may be deployed at any school with a system connection, and also collect the information from IoT products associated with IoT apps. Various types of IoT data could be directed to the correct place for more analysis based on performance demands. The higher goal information which has be resolved promptly could be prepared on fog/edge computing nodes, and they would be the closest to the IoT products which produce the information. The lower priority data, which isn't delay sensitive, could be directed to several aggregation nodes for more analysis and processing.

Besides the advantages which fog/edge computing is able to add, you will find a number of problems to incorporate fog/edge computing with IoT. A probable problem is how you can efficiently handle fog/edge computing infrastructure and allocate resources that are available to IoT products. Within every time, a big num ber of services may be requested by IoT systems, along with each fog/edge service node just has limited computing as well as storage capability. In this particular situation, most fog/edge nodes must be opti mally handled as well as allotted for IoT products (or maybe a pair of IoT products in a cluster) to offer requested services effectively. Another challenge is the best way to effectively manage fog/edge com puting resources. Even though the prior problem focuses on the user interface among fog/edge nodes as well as IoT services, this particular challenge focuses on the source management amongst fog/edge nodes.

When fog/edge nodes are issued to supply services, requirements that are different have being considered, such as program accessibility, energy usage, and also earnings. Consequently, exactly how to optically map the fog/edge service nodes to IoT products to encounter needs of IoT apps is still a compelling problem. In addition, privacy and security issues (authentication, access control, intrusion detection, loyalty management, etc.) contained fog/edge computing infrastructures which integrate with IoT stay furthermore tricky [48, 49, 79, 80]. The privacy and security problems may be mitigated by countermeasure technologies stated in Section V, and therefore the difficulties in resource allocation are reviewed below. The issues of materials allocation inside fog/edge computing based IoT may be divided as resources allocation involving end-devices as well as fog/edge node as well as information allocation amongst fog/edge nodes.

1) Resource Allocation Between End Devices and Fog/Edge Node: Because computing and saving resources are restricted to a fog/edge node, it's tough to entirely gratify all services requested by end users concurrently. In order to address this particular issue, each end-users could have a pleasure feature to evaluate the allocated resources to offer its requested service.

In a fog/edge computing based IoT, a selection of fog/edge nodes are attached, if a fog/edge node doesn't have sufficient assets to offer the requested services from close by end owners while its neighbouring nodes have extra online resources, the fog/edge node is able to go several neighbouring formation to its neighbouring nodes being prepared as well as stored information. By doing this, ser vices for the local end-users of its could be provided. This is connected to the resource allocation amongst fog/edge nodes, that will be discussed below.

2) Resource Allocation Between End Devices and Fog/Edge Node:

As the distributed architecture of fog/edge computing based IoT, most fog/edge nodes could be hooked up with one another through the system contacts and talk about the computing of theirs and saving information to supply service for end users. In this particular situation, if a fog/edge node doesn't have sufficient learning resource to supply neighbourhood requested services, the fog/edge node is able to go some requested services with lower goal amount to be processed within its neighbouring fog/edge nodes, which happen to have extra information. In the useful resource allocation amongst fog/edge nodes, a resource poor fog/edge node might not care about which resource-node allows it to offer computing services, along with a resource-rich node doesn't care about information from which it processes. The only person most fog/edge nodes care about is achieving the least price (minimum delay, etc.) in the complete fog/edge computing infrastructure. By taking the goal of the bare minimum delay.

In this section, an extensive overview of IoT has been provided, including architectures, enabling technologies and also the integration of fog/edge computing with IoT to allow for several applica. Especially, the connection as well as distinction between CPS and IoT happens to be clarified at the start. Architectures for IoT are discussed, like the SoA based four layer architecture. Additionally, to secure IoT, potential privacy and security issues which could influence the usefulness of IoT, and the potential solutions of theirs, are presented. In order to check out the fog/edge computing based IoT, the connection between IoT and fog/edge related issues and computing are talked about. Moreover, many uses, like the smart grid, wise transportation, and sensible cities, are presented showing just how fog/edge computing based IoT to be applied with real world programs. The primary reason for this particular area is providing a definite, comprehen sive, as well as strong knowledge of IoT and the integration of its with fog/edge computing, describing the breadth of subjects which IoT entails, and also highlighting regions that continue being unresolved, in an attempt to further promote the improvement of IoT.

2.2 Social Network

Social Network (SN) sites can be defined as a web-based service that facilitate human to create a profile online within a system for connecting with other users in the system for sharing virtual contents and connections with each other. Visibilities of the profiles, policies, terminilogies and nature of a system may vary from site to site [41].

Popularity of various SN sites imposes changes in the organization of online communities. Websites are focused on the communities of interests, however, SNs are focused on the human, not interests. Early open online networks e.g., Usenet and open discourse discussions were organized by subjects or as indicated by topical hierarchies, however, SNs are organized as personal (or ”egocentric”) networks with the person at the focal point of his/her own community. This more precisely reflects unmediated social structures, where ”the world is composed of networks, not groups” [189]. The evolve of social network sites have directed a resonant research context and our proposed SIoT framework is another contribution to the community.

In this thesis, we propose a framework where human and objects can be connected based on their interests. Moreover the interests are not limited only to the cyber world is extended to the physical world. Samrt societies can be formed using the proposed SIoT framework. In a smart society, not only the human but also the physical objects is considered as user. We detail the process and architecture in the Chapter 4.

2.3 Middleware for IoT

Middleware is a software program or maybe system programming which can offer an abstraction interposed between IoT technologies and applications [96, 202, 45, 197, 75, 72, 11, 74, 73, 76]. Within middleware, the specifics of various technologies are concealed, and the conventional interfaces are supplied to allow developers to concentrate on the improvement of uses without thinking about the compatibility between uses as well as infrastructures [17]. Consequently, by utilizing middleware, applications and devices with various interfaces can exchange share and information resources with one another.

Middleware has got the following advantages [17, 128], a) middleware is able to support different applications; b) middleware is able to operate on different OS's and platforms; c) middleware is able to support the distributed computing and also the interaction of services along with applications, devices, and heterogeneous networks; d) middleware is able to support regular protocols; along with e) middleware is able to offer regular interfaces, offering standard protocols and portability to allow interoperability, that enable middleware to play a a crucial role in standardization [47].

Middleware also can supply a healthy high-level interface for uses. With healthy interfaces, programs are able to work independently on the hardware as well as operating system. This feature makes middleware appropriate for IoT, because a lot of heterogeneous networks and devices are integrated, and these units and networks will be changed or updated frequently.

A number of research endeavors on middleware have been produced, and may be split into five groups [47, 155], including a) message oriented middleware; b) Semantic Web based middleware; c) location based service as well as surveillance middleware; d) correspondence middleware; and e) pervasive middleware. Especially, message-oriented middleware is able to offer the dependable info exchange along with communication protocols, and various platforms (e.g., AMQP, DDS, MQTT, and XMPP) [8, 47]. Semantic Web based middleware is able to offer the interactions as well as interoperability among different sensor networks. Instances of this particular group are the SoA based middleware [173], task computing based middleware [90], etc. Location-based service, as well as surveillance middleware, incorporate the locations of other information and devices to offer integrated benefit services [hundred nine]. Communication middleware is able to offer dependable communications among heterogeneous devices and applications. In interaction middleware, RFID based middleware (Fosstrak [1]), sensor network based mid-

[1] https://fosstrak.github.io/

dleware (TinyREST [132]) as well as the supervisory management as well as information acquisition are common examples. Pervasive middleware is created for the pervasive computing planet and offers services on heterogeneous and multiple platforms [162]. In order to integrate middleware directly into IoT, the following conflicts need to be addressed [47].

1) Interoperability challenge is connecting heterogeneous devices in communication and info exchange. 2) Scalability problem is to be successfully operated in either small-scale atmosphere or maybe large-scale setting which might entail an enormous quantity of items. 3) Abstraction provision problem is providing abstractions at different levels. 4) Spontaneous interaction problem is providing the dependable service for impulsive occasions. 5) In fixed infrastructure problem is providing dependable providers without requiring a fixed infrastructure. 6) Multiplicity challenge is supporting simultaneous communication among devices and also to select or schedule probably the most appropriate providers for devices by an enormous set of services. The middleware for IoT must achieve trust, privacy, and security. In next two sections, we present a bunch of requirements for a middleware to support IoT. The requirements are derived according to the attributes of IoT's infrastructure.

2.3.1 Middleware Service Constraints

The IoT middleware's requirements are categorized as both functional and nonfunctional categories. Services or funtions are fall into functional requiremnts e.g., abstractions andresource management, QoS support or performance issues are fall into nonfuntional requiremnts. We do not consider any domain specific middleware rather we consider generic midleware that can provide services to multiple different applications [156].

- *Resource Discovery:* Heterogeneous devies and objects are connected through IoT infrastructure and considered as IoT resources. They differ on power mechanism, memory, capacity, physical size, communication mechanism and provided service. Due to the dynamicity of IoT infrastructure and the dynamicity and hugeness of IoT resources, it is not practical to consider human intervention for resource discovery. As a result resource discover needs to be an automated process. The primary requirement is the IoT objects should announce their presence along with their capabilities and hardware resources even when there is no available network. However, in the centralized distributed systems, resource publications, discovery and communication are normally managed by a dedicated server. Discovery mechanisms additionally have to scale well, and there ought to be the effective division of discovery load, because of the IoT'resource-constrained source constrained products.

- *Resource Management:* A suitable QoS is expected for most apps, and also in an atmosphere where materials which affect QoS are constrained, such as the IoT, it's crucial that users are supplied with a program which manages the materials. This means the source used must be administered, resources allocated or maybe provision in a good fashion, as well as source disputes solved. Inside IoT architectures, particularly in service-oriented or perhaps virtual machine (VM) based architectures, middleware must facilitate likely spontaneous a useful resource (service) (re)composition, to satisfy the application requirements.

- *Data Management:* Data are crucial in IoT applications. In the IoT, data means generally sensed information or perhaps any network infrastructure information of interest to applications. An IoT middleware must offer data management solutions to programs, like information acquisition, information processing (which includes preprocessing), and information storage space. Preprocessing can include data filtering, data aggregation and data compression.

- *Event Management:* There are likely an enormous amount of incidents produced within IoT programs, that ought to be managed as a fundamental component of an IoT middleware. Event management transforms just noticed functions to substantial occasions. It must offer a real-time evaluation of high-velocity information to ensure that downstream programs are pushed by correct, intelligence, and real-time information.

- *Code Management:*Deploying code within an IoT environment is difficult and also must be exclusively backed through the middleware. Particularly, code allocation and code migration solutions are needed. Code allocation selects the pair of equipment or maybe sensor nodes to be utilized to complete an end user or even the application level process. Code migration transfers one particular node/device's code to a different one, possibly reprogramming nodes within the system. The code is easily transportable and code migration service is also need to be available that loads the data computation.

  Main *nonfunctioanl* requirements of IoT middleware are given below,

- *Scalability:* An IoT middleware must be scalable to cater to development within the IoT's network as well as applications/ expertise. Thinking about the dimensions on the IoT's network, IPv6 is a really scalable alternative for addressability, as it can easily cope with a substantial amount of items which have to become incorporated inside the IoT [sixty eight]. Unfastened coupling and also virtualization within middleware is helpful for boosting scalability, particularly software as well as assistance amount scalability, by hiding the intricacy on the basic hardware or even system reason as well as setup.

- *Real time or timeliness:* A middleware should offer real-time expertise if the correctness of a function that supports not just rely on its logical correctness but additionally on time whereby it's performed. As the IoT will contend with lots of real-time uses e.g., transportation, healthcare, monitoring, on-time shipping and delivery information. Delayed information may cause the services useless.

- *Reliability:* A middleware must stay functional for the length of a quest, maybe even within the existence of errors. The middleware's dependability eventually aids within obtaining systems level dependability. Each entity and services need to be dependable to obtain overall reliability. Middleware must be dependable to attain general reliability, including interaction, data technologies, plus products of all the layers.

- *Availability:* Even when there's a failure in the systems, the fixing period and the frequency of failure should be sufficiently small to accomplish the preferred availability. Highest fault tolerance mechanism is required to ensure the desired reliability and availability.

- *Privacy and Security:* Privacy as well as security: Security is crucial towards the functioning of IoT. In the IoT middleware, security has to be considered when it comes to all the nonfunctional and functional blocks including the user profile management. Context-awareness of middleware could disclose private information (e.g., the location of an item or perhaps a person). The users and items privacy must be ensured at each level of a middleware.

- *Ease of deployment:* The updates of middleware should be deployable by anyone without requiring the expert knowledge.

2.3.2 Architectural Requiremnts

The architectural requirements incorporated within this section are fashioned to help program designers. They contain demands for programming abstractions along with other implementation level considerations.

- *Programming abstraction:* Supplying an API for application designers is a significant practical necessity for just about any middleware. When it comes to the application program or even system developer, high-level programming interfaces have to isolate the development of the applications or perhaps services coming from the heterogeneous IoT operations. When designing an API, the degree of abstraction, the programming paradigm and the type of interface all need to be considered. The degree of abstraction refers to the way the application program developer views the system. The programming paradigm (e.g., publish/subscribe) relates to the type for building or perhaps programming the applications or even providers. The type of interface describes the model of the programming interface. For example, descriptive interfaces provide SQL like languages for information query [2], XML based specification documents for context setup [160].

- *Interoperability:* A middleware must allow heterogeneous devices/technologies/applications, with no extra work out of the application program or even a service developer. Heterogeneous components should be in a position to swap data and services. Interoperability inside a middleware can be seen as a result of the system, syntactic, along with semantic perspectives, all of that should be catered for in an IoT. Semantic interoperability refers towards the significance of information and/or perhaps a service, and really should permit interchange in between the transforming and evergrowing range of providers and equipment in IoT. Substantial information regarding services will likely be helpful for the owners within creating several services as semantic data could be much better understood by "things" or "objects" as well as human as compared to conventional protocol explanations [106, 25].

- *Service-Based:* A middleware architecture must be service-based to provide higher convenience when advanced and new tasks are to be added to the IoT's middleware. Abstraction is provided by a service based middleware for the complicated underlying hardware by way of a set of services (e.g., reliability, data management, security) desired by applications. These along with other complex services created,

[2]TinyDB: http://telegraph.cs.berkeley.edu/tinydb

implemented, along with incorporated in a service-based framework to provide an adaptable and simple environment for application development.

- *Adaptive:* A middleware has to be adaptive so that it is able to evolve to accommodate itself into modifications in its circumstances or perhaps environment. In IoT, network environment, application requirements and context asre dynamic and are likely to change change frequently. In order to guarantee user satisfaction and effectiveness of the IoT, a middleware must dynamically adjust and adapt to place almost all types of variations.

- *Context-aware:*Context-awareness is a vital requirement in generating adaptive methods as well as in creating value from sensed data. The IoT's middleware architecture needs to be concerned about the context of users, devices, environment to offer desired and suitable services to users.

- *Autonomous:* It indicates self-governed, IoT objects and applications are active participants in this procedure and so they must be enabled to interact and talk among themselves without immediate human intervention [92, 186]. Use of intelligence including autonomous agents, embedded intelligence [95], predictive, along with practical approaches (e.g., a prediction engine) in middleware is able to fulfil this particular requirement [140, 156].

- *Distributed:* A large-scale IoT system's applications, devices and users exchange information and collaborate with one another. They are likely to be geographically distributed, therefore a centralized view or alternatively, middleware implementation won't be adequate to help numerous distributed applications or services. A middleware implementation must support functions which are sent out across the physical infrastructure of the IoT.

2.4 QoS based Web Service Composition

Web Service is identified as a software program structure meant to help interoperable machine-to-machine interaction and yes it may be invoked, located, and published through the Web. A great deal of researches has centred on the procedure of Web Service Composition. Despite the functional qualities in equivalent program organizations, considerable interest was drawn to the non functional features, particularly Quality of Service (QoS). In fact, the QoS standpoint is viewed as probably the most important essential

Categories	QoS Criteria	Definition
Task-based	Price/Cost	The price that a service requester has to pay for invoking the service
Task-based	Response time	The time interval between the moment when a service is invoked and the moment when it is finished
Performance-based	Reliability	The probability that a request is correctly responded within the maximum expected time
Performance-based	Availability	The probability that a service is available
User-based	Reputation	The average ranking given to the service by end users according to their own experiences

Table 2.1 QoS criteria for web service [171]

problem for service composition since it may often figure out the last overall performance of the composed program class and it is directly associated to user satisfaction.

In this section we mainly discuss the QoS based service composition, more specifically the service selection mechanism.

2.4.1 QoS Models

The idea of workflow and its application for Web service composition was initially suggested by Cardoso[1]. As Cardoso discussed, the workflow is generally made up of task plus transitions which denote the dependencies between task and are connected with an allowing probability.

According to several scientific studies, the elemental components for workflow design may include loop, choice, parallel, and sequence. At the first scientific studies on the end-to-end web service composition, different quality key elements are recommended. Zeng et al. [205] establish the QoS attributes as the execution price, execution duration, reputation, reliability and availability. Cardoso et al. [44] concentrated on three metrics of the task period, task reliability as well as task cost. Response time, service cost and service availability and service reliability were considered in T. Yu et al.'s [3] work.

As Table 2.1 shows the task-based criteria are interested in the particular request that the service gets and their values are often depending on the Web service description, even though the others are definitely more linked together with the service's overall performance in previous times. Particularly, the performance-based requirements could be exclusively calculated by utilizing the independent information captured by each execution, and also the user based criteria is much more comprehensive since it requires feedback from various people.

As compared to the various other characteristics, the task-based requirements are much more well known for the present investigations. Based on Cardoso [44], the entire price when the service is invoked, will be the amount of enactment cost(EC) and realization cost(RC), with the former associated with the management as well as monitoring of the service, and also the latter representing the price for the runtime delivery. Likewise, the reaction time is also made up of procedure time (PT) and also delay time (DT) that describes the non-value-added time like the queuing delay as well as the assembly time.

2.4.2 Mathematical Approaches for Web Service Composition

The QoS aware service composition issue has been described as mathematical problems e.g., Integer Linear Programming (ILP) (aka IP), Single Objective Problem with QoS constraints and Multiple Objective Problem with QoS constraints would be the most typical ones.

ILP [204, 212] is able to help the service composition to an effective option without producing all of the possible composite services[8]. Nevertheless, it is only available for the composition problems with modest volumes since the conventional Branch and Bound method has the computation limits. However, it looks for the linearization of the objective function and corresponding constraints.

The single objective problems [100, 101] believe that the weights for QoS characteristics are provided, but the transformation from multi-objective to single-objective generally can't be attained by merely weighted summation. The many unbiased issues have no demands for all the weights of QoS details, and the unbiased feature for every QoS parameter is combined generating a number of Pareto optimal solutions that can be selected by owners in conditions of the own personal preferences of theirs. As compared to the former, the latter appears to be far more adaptable which enable them to be much better put on to service composition.

The usual mathematical option for the composition issue was suggested by Zeng et al.[204, 205], that continues to be used by relevant scientists. Methods for equally local selection and global planning was reviewed. The local choice is dependent on Multiple Criteria Decision Making (MCDM) and Simple Additive Weighting (SAW) working with the evaluation of individual web service while the global one mostly includes three actions, the dedication of one execution path, the comparison between several delivery paths and also the work of Integer Programming (IP).

2.4.3 Heuristic Algorithmic Approaches for Web Service Composition

QoS-aware service composition is NP hard, plus it is usually described as a Multi dimension Multi-choice Knapsack Problem (MMKP) that searches for the composition which provides the greatest total fitness when fulfilling QoS constraints. It generally consists of numerous constraints as well as can't succeed when a lot of services are engaged. The unethical approach is utilized to assist selecting the ideal choice service ideal for the execution[181]. Nevertheless, it is in the hands of the local strategy that could just get permission to access the rough answer instead of the suitable optimal one. Thus, the heuristic algorithms are getting more prominent for QoS aware service composition, among that the genetic algorithm (GA) is reported to be probably the most pervasive one.

GA [123] is dependent on the concept of Darwinian evolution plus it generally has businesses of crossover, mutation and selection operations. By gradual evolution, probably the most adaptable chromosome may be attained. This way, GA may nicely react to the optimizing process and also the last chromosome is comparable to the most effective answer. For QoS aware service composition, the chromosome belongs to a specific composition plan and depict the corresponding abstract providers. Based on the QoS values for various solutions, the physical fitness feature may be derived.

The primary difficulty for GA would be that the conclusion based on this particular algorithm might be local optimum which is going to lead to the pragmatic barrier of its down the road. And Compared to IP, GA is much more effective just if the amount of pertinent solutions are large [82]. Many strategies have transpired. Liu et al. [81] converted the NP-hard issue in to a multi objective optimization problem along with constraints and proposed GODSS to understand the service selection with QoS global optimization. The Elitist Selection Genetic Algorithm (ESGA) is provided by Dong et al. [64], the place that the integer encoding is used as the encoding principle and also the original population selection tactic guarantees the status of selected service couldn't be even worse compared to the general fitness.

The style of the coding pattern of chromosome, , evolution operations and fitness feature and selection mechanism has an immediate impact on the effectiveness as well as worldwide astringency[124]. Zhang et al. [124] has suggested the binary strings of chromosome, wherein every gene in chromosome represented a program prospect with values zero as well as one. The one dimension coding pattern of any chromosome was presented for describing the program structure by Canfora et al. [81]. Since the many paths and service replanning issue was generally not used into the bank account in a single dimension coding system, the relation matrix coding program was proposed [124, 210, 209]. Zhang et al. [124] has additionally offered a brand new evolution at the performance of the population along with the population policy depending on the blend of population variety and simulated annealing that will converge asymptotically to the perfect option in a nearby area and it is proved to become an effective stochastic optimization method.

2.5 Object Virtualization

In this section, we describe the present status of IoT-object virtualization also known as Virtual Object (VO), its definition, characteristics and functionalities.

2.5.1 Virtual Objects Features and Definitions

Several definitions are available for the "virtual object" in IoT paradigm. The main reason now for the vagueness around this term is a consequence of the outstretched use of the term "virtual" in several contexts since the 70s, which range from the potential for providing an effective facsimile of one or even more full computer systems in one device [88, 87] to the capability to provide an adaptable abstraction of entities like time and memory [105, 121]. The 90s had been the era of active virtual environments and also for at first chance, the terms virtual and object were linked together, so that a virtual object was described as a digital representation of the functionalities as well as the shape of real-world objects through what human can communicate with it [68]. However, in this view, there's not any true possibility to have interaction with an actual object as everything is only virtual.

Objects virtualisation in the IoT started with the very first associated solutions, i.e. RIFD tags [185, 70]. Even with the simplicity of theirs and passive RFID-based identification, limitations systems made it possible for the implementation of a broad range of novel uses by bridging the gap in between the actual physical world (i.e., tagged real-world objects) as well as the virtual world (i.e., application software). For instance, the virtual objects proposed by Barrett et al. [33] enable to improve labelled actual physical objects with electronic info. In a comparable manner, Want et al. [187] report a range of scenarios in which virtual objects are utilized to augment common objects through embedded RFID tags. Finally, but not least, the Cooltown task [108] attributes a page not just to places and people but additionally to arbitrary items. With respect on the meaning of a virtual item in the virtual environment described in the 90s, in the IoT the virtual object should have a version in the actual physical world, of which it exposes the services, and also doesn't have always a shape. This characterization will remain basically unchanged in time though the virtual objects will develop brand new qualities.

The digital counterparts analysed in the first IoT exposed just one particular functionality to the final customers based on the specific program at hand. Among the dimensions which characterise the meaning of a virtual item may be the connection between actual items, i.e. the services of theirs, and the virtual items themselves: for instance, a smartphone might expose all the services of its through an individual virtual object or maybe it might have distinct virtual objects dependent where solutions are made available, e.g. 1 for the localisation services plus 1 for the temperature sensing; in the exact same way, it's possible to utilize a single virtual object to collect info on exactly the same service from many actual physical objects. Clearly, every option carries its own disadvantages and advantages in conditions of addressing, service finding or maybe source reusability. This particular dimension has been addressed differently after a while, as mentioned in the following.

The pioneering work of Langheinrich et al. [114] again to 2000 is the very first which addresses demands for a big scale deployment of RFID tags. From the paper of theirs, whether or not the definition of virtual objects stays the same, they get a brand new characteristic: an identification and addressing pattern to be able to have the ability to find the virtual items on the web. In this particular proposal, certain virtual objects, like emails and Websites, couldn't be associated with an actual object. Furthermore, the authors consider the potential for having just one virtual object for an actual physical object (if any), consequently resulting in a one (or less)-to-one association.

An additional action in the evolution of the virtual item (or maybe virtual counterpart) is offered in the job of Römer et al.[157]. The authors introduce the idea of meta counterparts, which presents an entire range of actual physical items. Because of this, it is possible to handle the virtualisation better than having a distributed implementation, since it allows in order to reduce the resource consumption and then to improve the service discovery. This presents the very first effort to connect the virtual objects based on the functionalities of theirs: every virtual object will be able to manage not just the set of services associated with an individual true object but also the products of a number of the same items after which it moves to some many-to-one the connection between virtual and real objects.

Within the last five years, many research projects were founded to propose a structure for the IoT, resulting in a further evolution of the functionalities of the virtual object.

The CONVERGENCE project [3], for instance, makes use of Versatile Digital Item (VDI), a typical box for all kinds of , including one or even more physical objects as well as the corresponding metadata. The proposed virtual object is akin to the camera offered in [18] with a many-to-one association between real VDI and objects, i.e. a one VDI can represent a couple of digital contents. Nevertheless, virtual objects are not only digital counterparts of real-world objects but also now supply a semantic enrichment of the acquired information, which helps make the discovery of services much easier, since metadata is consumed to index the virtual items.

Some other two intriguing illustrations of the evolution of the virtual items are SENSEI [4] [153] and IoT-A [5]. The virtual objects, called resources within SENSEI along with virtual entities in IoT-A, become conscious of the context where the physical an item operates then acquire the ability to improve the information received by the real-life objects with environmental info. Additionally, these tasks propose for the very first time methods to orchestrate IoT services to be able to blend together several virtual objects and also give high-level services to the user or even to the application. Nevertheless, IoT-A and SENSEI differ for the association between virtual and real items. SENSEI resources could be connected with one or even more physical objects, therefore providing the very same kind of association of CONVERGENCE. On the other hand, the IoT-A actual physical item is decomposed in its simple services, therefore, providing initially a one-to-many connection with the virtual entities.

An additional step in the evolution of the virtual item is supplied by the COMPOSE [10] as well as the iCore [6] tasks. Virtual objects get a cognitive ability, allowing them to use the collected info to create a pertinent choice and acting upon it. Additionally, even if COMPOSE uses the identical kind of association of IoT-A, a fascinating trait of the iCore task is which not merely a virtual object is usually connected with one (or maybe more)real-world items but additionally a real-world object to one plus virtual objects, consequently resulting in a many-to-many association.

[3]http://www.ict-convergence.eu/
[4]http://www.ict-sensei.org/
[5]http://www.iot-a.eu/
[6]http://ai-group.ds.unipi.gr/ai-group/node/2207

According to these proposals, it occurs the virtual items have become the essential element to deal with some crucial issues in the IoT, whose main ones are: fast deployment of new network architectures and elements by linking the virtual object to outside services; allowing for the coexistence of heterogeneous network architectures over the same infrastructure; providing all of the time reachability, flat when the physical devices is temporarily unavailable; bring intelligence on the object to be able to learn the context and implement self-management operates. All of the operations done by the virtual objects to offer these functionalities are thoroughly talked about in the next portion, highlighting the primary key value.

In the following, we are going to refer to a virtual object/entity/ counterpart as terms that are interchangeable, with the significance of an electronic representation of the real world object.

2.5.2 Purpose of Virtual Objects

The intention of this section is describing the functionalities that are given to the VO in the IoT area.

The primary idea of the IoT is definitely the pervasive presence of connected objects [16]. Because the birth of its, the idea of IoT has turned into numerous dimensions, encompassing different solutions able to offer real-world intelligence as well as goal oriented effort of distributed smart objects via global networks or local interconnections. As outlined in [215], the present circumstance continues to be represented by a mixture of many "Intranets" of Things, that are changing into a more integrated and heterogeneous phone system to converge on the melting pot which characterises the IoT ecosystem. Such a complex atmosphere is confronted with a varied and long list of issues that are different. Our investigation provides following issues that need to be addressed are given below.

- Heterogeneity
- Scalability
- Identification
- Plug and Play
- Search and Discovery
- Constrained Resources

- Quality Management
- Mobility
- Security and Privacy

The most difficult struggle of the IoT would be to have the ability to handle the deployment of uses concerning heterogeneous objects, frequently shifting in environments that are big, in a manner that fulfils the quality needs of the application program itself, while not overloading the system online resources. The virtualization of objects appears to be the best answer for this problem, as it's intended to support:

- quick deployment of new network components and architectures [50];
- co-existence of heterogeneous community architectures over the same infrastructure [143];
- all of the time reachability [51];
- self-management of network objects, achieved by context awareness [151].

In the following, we analyse the primary key functionalities that're supplied by virtualization layers to fulfil the IoT challenges discussed in Section III A (Figure two visually displays the relation between functionalities and issues). We distinguish between virtualization specific functionalities, that are created by the virtualization layer, along with virtualization enhanced functionalities, that are functionalities which currently exist in IoT architectures, though the performance of theirs is enhanced by the usage of any virtualization level.

1) Virtualization: Specific Functionalities *a) Semantic description* Just about the most crucial demands of the IoT is the fact that of creating heterogeneous objects plug-and-playable as well as interoperable. The moment an item joins a system, it must be right away supplied with mechanisms which allow interaction with the external planet. Community find mechanisms are accustomed to dynamically learn as well as configure brand new virtual objects at runtime. Probably the most powerful as well as efficient resolution to represent IoT items is by using semantic technologies.

The outcome of the semantic description of an item, or maybe a team of items, will be the virtual object version. The virtual item design contains objects' characteristics;

objects' location; resources, services as well as quality parameters supplied by objects. Whenever a brand new object is detected, it's associated with a brand new virtual object instantiation [1].

The very first efforts of standardisation wearing semantic explanation came out of the Wireless Sensor Network (WSN) area. The W3C's Semantic Sensor Networks (SSN) ontology [55], moreover the OGC Sensor Web Enablement collection [55] of XML based requirements are simply a number of good examples of semantic description languages. These're currently utilized in numerous virtualization architectures e.g., SENSEI [153] and IoT-A [7].

Not merely is semantic description able in order to deal with heterogeneity also to offer interoperability amongst items, though it's likewise extremely effective in supporting search and find operations [27]. Certainly, search and find mechanisms are typically utilized to identify the virtual objects which are best suited to do a certain application 's job (or maybe a part of that job) [3]. At this particular objective, online search engine are used, which are often based on semantic search technologies.

b) Context awareness: This feature will be the power to obtain, analyse as well as understand info regarding the ecosystem. Recognition is necessary to streamline the finding of info regarding the object itself if this info: can't easily be achieved, isn't created explicit in the required format, and requires aggregation along with other info sources before it may be utilized [154]. Discovered info is represented by resources as well as services which are made ideal by some other nodes, and also information gathered by them. Even though context recognition may be done in regular IoT by sensible objects, in virtualization levels this function is given to virtual items. This way, recognition is made it possible for maybe even in items with restricted capabilities: smarter activities which they wouldn't have the ability to do are supplied by the virtualization layer, which allows recognition mechanisms. Consequently, it could be considered a virtualization specific functionality.

c) Cognitive management: This virtualization specific function is absolutely regarding context awareness: the data acquired all about the earth where the virtual object is running is utilized to create related choices as well as act upon them [107]. Each time a virtual object perceives an alteration of the context, it processes info with the use of predictive models and optimisation algorithms [77, 170, 184]. Based on the obtained outcomes, the virtual item autonomously reconfigures the item it refers to be able to adjust the behaviour of it's to improvements.

[7]http://www.iot-a.eu/

The cognitive management function is very crucial for a lot of areas [1, 77, 170, 184].

Instantly scheduling and detection of incidents by the objects, that also react by performing a self-configuration of the functioning parameters of theirs.

Source consumption is optimised (e.g. energy, storage), communication bandwidth, processing capabilities. Optimisation algorithms are applied to effectively allocate resources to projects, so long as needed quality circumstances continue to be happy.

Fault management is performed anytime the virtual item can't speak with the item it's related with. In this particular situation, the virtual object: estimates the object declare working with cognitive/prediction methods; proposes substitute backlinks whereby information could be forwarded.

2) Virtualization: Enhanced Functionalities: *a) Addressing and naming:* The moment an object is created on a system, it's essential that address and/or a name is given to it. This's a virtualization-enhanced functionality, as in conventional IoT it's in control of addressing true objects. In a virtualization layer, it provides names and addresses to recently created virtual objects, so that the resources of theirs, services as well as information are accessible by every other entity on the Internet [38], in a manner that scales nicely once the system colour expands. The location-dependent addressing system utilized in the present Internet architecture [38] isn't appropriate together with the contemporary perspective of IoT, that is based on sharing information. Thus, information-centric architectures are claimed to function as the future of IoT. These architectures are especially appealing in virtual object based architectures, because of the inclination of theirs towards semantic search mechanisms. Because of this, naming is a vital component for virtual objects' identification as well as interaction [1]. discovery and

b) Search and Discovery: The dynamics of IoT is the fact that of a dynamic system whereby items constantly subscribe, move and then leave the platform. As a result, a virtualization layer will be able to manage virtual objects which degrade, vanish or perhaps re-appear.

In this particular context, automated find and hunt mechanisms are required to attain an accurate and scalable interoperability among virtual objects [26]. This particular performance is virtualization-enhanced: it's the virtual objects' counterpart of search as well as finds done on genuine items. Since virtual objects enable the systems which expose the attributes as well as abilities of the actual physical items they virtualise, the search as well as finding procedures determine which virtual items supply the materials as well as services which best fit the quality level needed by the software given to the IoT phone. As released in Section III B1a, the semantic explanation item eases the discovery of virtual items. Because of semantic engines, the usage of the same language enable research among products with very unique qualities, e.g. a smartphone and a sensor may certainly react to the request of realizing temperature. Moreover, the extended use of the cloud in virtualization architectures produces discovery and search much faster, actually achieving real-time search [24].

Particularly, Peer-to-Peer (P2P) find devices, that are influenced by information-centric technologies, are better than various other find mechanisms for the capacity of theirs of utilizing semantic info to find out information and services [38]. In reality, P2P solutions use a distributed infrastructure, over what service and resources information is dispersed and also managed [62]. Using semantic based algorithms, information that is relevant is retrieved in a distributed way. Not merely are scalability as well as accessibility enhanced, but additionally achieved search engine results are definitely more accurate with regard to other mechanisms.

c) Mobility control: Mobility control isn't a negligible problem for virtual objects: when an item moves from a single area to the next, the virtual counterpart of it should take notice of the migration of its. This task may result very hard, especially once the object becomes unavailable for extended periods of time because of not enough connectivity. In these instances, the cognitive capacities of the virtual item are incredibly important: it's required to be resilient to context modifications, by attempting to keep the operational functionalities with a suitable quality amount [1]. This additional implies trying to keep a high-quality communication link with its similar object anytime possible, definitely taking advantage of opportunistic and ultimately intermittent connections. Lastly, since the virtual item might be storing from date info of the item, a parameter indicating just how much time is transferred out of the final time the virtual item received information from the physical counterpart of its is needed. In case needed, the virtual item will be able to work with predictive types on this information, to calculate the state of its connected object.

d) Accounting plus authentication: Accounting and authentication IoT function is improved by virtual objects, and they explain the users/entities whose entry is authorised, in addition to the owner of theirs or creator. The access of theirs has to be secured through the usage of encryption keys [1].

Concerning data security, virtual items are able to create semantics to be able to create a universal trust management method [180]. In such a method, autonomous trust management process as well as modules progression and interpret trust information, and also offer explicit meaning to believe in information applying semantic annotation.

2.6 Summary and Discussion

In this chapter, we have presented the background technologies we have considered in this thesis. IoT is the main area we have focused in this research. We have surveyed the present status, past story and future direction of IoT including the enabling technologies. We have also discussed the Semantic web and related technologies and their relation and with IoT.

Chapter 3

State-of-the-Art

In this chapter, we present the state-of-the art. In particular, it describes the relevant technologies, methods and their current status.

3.1 Semantic Stream Data Processing

It's very intriguing to discover the way the World wide Web (WWW), that is an invention of Sir Tim Berners Lee just twenty-eight years back, has transformed the lives of individuals in these kinds of a deep way. Statistics [1] indicate that since the very first ever website [2] was released, nowadays there tend to be more than 9.38×109 sites, so the quantity of Internet users has reached to much more than three billion, approximately 40% of the entire population on Earth. In 2001, Sir Tim Berners Lee described his vision on the coming generation of WWW, considered the Semantic Web [35]. The Semantic Web (SW) is all about marketing expertise, rather than documents, as in the standard WWW. The same as working with hypertext files to abstract from physical as well as network levels of the online world, the Semantic Web pick machine accessible expertise to abstract from net files as well as uses. Different Semantic Web techniques are utilized in this thesis. In the following, the fundamental ideas in the Semantic Web are released, the fundamental idea of creating connected information is provided and also the latest program of knowledge representation for sensor products are talked about.

[1] http://www.internetlivestats.com
[2] http://info.cern.ch

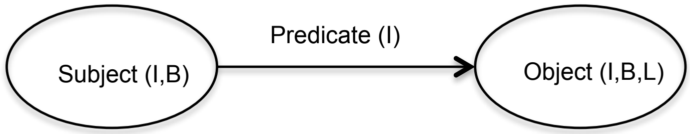

Fig. 3.1 Simple representation of a RDF triple

3.1.1 Overview of Semantic Web

For Semantic Web, knowledge is encoded as ontologies in the type of interlinking graphs, in which nodes stand for ontological terms as well as principles as well as edges symbolize relations. An ontology is often a knowledge base applied as a graph website. Originating from a specialized point of view, an assortment of Semantic Web methods as well as standards are created to model, represent as well as query the info offered by those graph databases. Along with these strategies, 3 specialized requirements would be the foundations of the Semantic Web: Resource Description Framework (RDF) [3] , Web Ontology Language4 (OWL) [4] along with Simple RDF and Protocol Query Language [5] (SPARQL).

Resource Description Framework (RDF) RDF specifies the data model for the Semantic Web. RDF describes the information on the Web using triples consisting of sub- jects, predicates and objects. Figure 3.1 shows a graphical representation of a triple, where the subject and object are represented as ovals and the predicate as a directed link. An RDF triple states that the relation represented by the predicate exists between the subject and object, hence it is also known as an RDF statement. A set of RDF triples constitutes an RDF graph. Resources in RDF graphs are iden- tifiable by Internationalised Resource Identifiers (IRIs), which are Unicode strings. Literals in RDF graphs are used to describe concrete data values, which have data types, e.g., integers, strings and float values. Apart from IRIs and Literals, blank nodes are also used in RDF graphs to indicate that something exists, without identifying specific resources.

RDF Schema (RDFS) [6] offers an extension to simple RDF vocabulary, allowing defining taxonomies (i.e., category or maybe home hierarchies), home domains as well as ranges, and also pots etcetera. Semantics are supplied to RDFS and RDF to ensure that a device can't just understand the ontology specified making use of the vocabularies but additionally infer or even entail implicit RDF claims from explicitly specified ones.

[3] RDF 1.1 concepts and abstract syntax: http://www.w3.org/TR/rdf11-concepts/.
[4] OWL language overview: http://www.w3.org/TR/owl-features/
[5] SPARQL 1.1 standard: http://www.w3.org/TR/sparql11-overview/
[6] RDFS 1.1: http://www.w3.org/TR/rdf-schema/

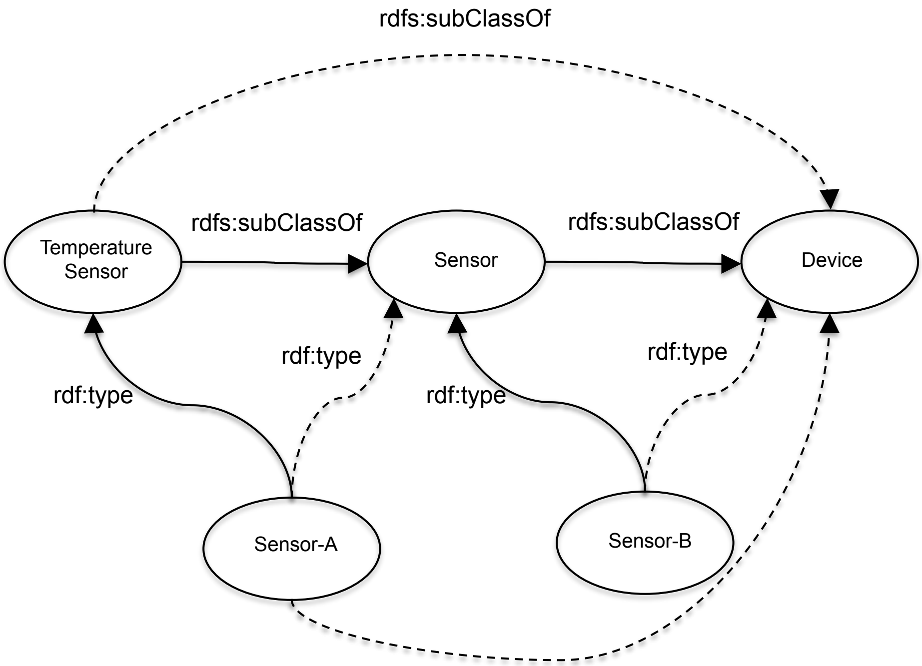

Fig. 3.2 RDFS inference

Web Ontology Language OWL OWL is created to become knowledge representation language just for the Semantic Web. Additional vocabulary is provided by it and the proper semantics to RDF and RDF Schema. In doing this, OWL offers higher expressive ness and much better machine interoperability when utilized to describe ontologies. OWL has 3 sub languages: OWL Lite, OWL DL and OWL Full. As the labels suggest, OWL Lite provides the lowest expressiveness as well as OWL Full contains the highest.

Simple Protocol and RDF Query Language SPARQL SPARQL offers the vocabulary as well as protocol for querying RDF graphs. It works on a syntax akin to SQL. Listing 3.1.1 gives a fundamental SPARQL query illustration with the graph shown in Figure 3.2. The execution results from this query is dependent upon what sort of inferencing as well as reasoning assistance is supplied by the SPARQL motor implementation. A SPARQL query engine without reason assistance will just provide Sensor one when the result, while a query engine with RDFS amount reasoning support is going to provide each Sensor one and Sensor two as outcomes.

```
Select ?sensorId Where {
?sensorType rdfs:subClassOf :Device.
?sensorId rdt:type ?sensorType.
}
```

Listing 1 Sample of a simple SPARQL query

Overview of Linked Data

• Assign URIs to the entities. Published entities needs to have the URIs of theirs which chart over HTTP protocol to the RDF representation of theirs. For instance, each sensor must have a distinctive URI, that links to the information of its in RDF. • Provide metadata regarding published information. Published information must be discussed by the way of metadata to boost the usefulness of its for information customers. Data should contain info on the creator of its, creation method and creation date. Publishers must also offer alternative means for accessing the data of theirs. During the last years, an increasing adoption of Linked An explosion and data principles of datasets specified in RDF is noticed. Original adopters included primarily academic researchers and developers. Nevertheless, more lately there's a significant interest from organisations in creating the data of theirs within RDF. Several of the most visible examples include BBC music data [7], British authorities data [8] or maybe Library of Congress data [9]. At exactly the same period, an increasing amount of public vocabularies (ontologies) as well as the interconnectedness of theirs are made, which forms a Linked Data Cloud [10].

3.1.2 RDF Stream Processing Engines

Stream data processing refers to an continuous processing of unbounded, continuous and time stamped data. Streams are generated through numerous modalities and different levels of granularity; e.g., IoT objects, social networks, web, financial market and more. These diverse stream data need to be queried and processed to retrieve information to satisfy application requirements. However, traditional DBMS is not suitable to process stream data on-the-fly because of their store and pull based strategy where data freshness is compromised. As a result stream processing systems are considered in this thesis to processing data on-the-fly as real-time processing.

Over the last few years, several RDF stream processing (RSP) engines have been proposed for efficient processing of RDF streams [31, 117, 37, 14, 111]. However few of them are still on their cradle and are not published with the deployable resources. Hence, the most popular and deployable RSP engines are. CSPARQL, CQELS has been considered in this thesis.

[7]British Broadcasting Company: http://www.bbc.co.uk
[8]British government: http://data.gov.uk, last accessed
[9]Library of Congress: http://id.loc.gov, last accessed
[10]LoD cloud: http://lod-cloud.net, last accessed

C-SPARQL (Continuous SPARQL) [31] is one of the first contributions in the form of new query language over streams of RDF data. It aims to fill the gap between DSMS (Dynamic Stream Management Systems) and SPARQL [5,18]. C-SPARQL supports timestamped RDF triples, continuous queries over streams and a periodic execution strategy. In C-SPARQL a time window can be a sliding or a tumbling mode, focusing on the most recent items in the stream. Determining the right window size and execution method can have a great impact on output results of stream query processors. Work in [116] has shown that C-SPARQL suffers from duplicate results for simple queries and misses some certain output in complex queries. This is most probably due to implementing the Rstream operator as a Relation-to-Stream operator in which old triples do not get removed from the time window. However, other evaluation results in [208, 61] have shown that C-SPARQL performs better among other RSP engines with different benchmark queries. This diversity in output result is true for other processors as well. It is evident that the performance of existing systems can be seriously affected by a certain set of queries and scenarios. We believe a adaptive solution can best serve the changing requirements and conditions for RSP.

CQELS [117] is a more recent language accompanied with an engine to process RSP and static data. CQELS also borrowed the processing model of classical DSMS and implements the window concept and relational operators like C-SPARQL. However, they differ from each other in the execution method where CQELS follows the data driven strategy and C-SPARQL follows the time driven strategy. The data driven strategy is a continuous monitoring to fetch data. This might be infeasible in situations where streams have high arrival rates and also it consumes more resources as it has to wait for the data arrive all the time. The time driven execution strategy executes the query periodically [117], it is a trigger based approach where the result is triggered after a certain time. The disadvantage of periodic query is that results may be stale if the re-executions [117, 116] frequency is lower than the frequency of the updates. A continuous query contains sliding windows to deal with unbounded nature of data streams. When the window slides, the query is continuously computed to reflect both new tuples entering the windows and old tuples expiring from the window. Based on this two opposite philosophies, the two execution approaches have a large impact on the output results [208, 61, 116] . We are considering this fact to improve existing stream processing solutions.

In most of the RSP engines, different combinations of R2S (Relation-to- Stream) are available, but we believe that a combination of all R2S [13] operators and features can produce better performance if the system is able to select the best operator based on the data and applications requirement. Currently, not a single approach exists which can address the above-mentioned issues within a single solution. Hence we introduced an

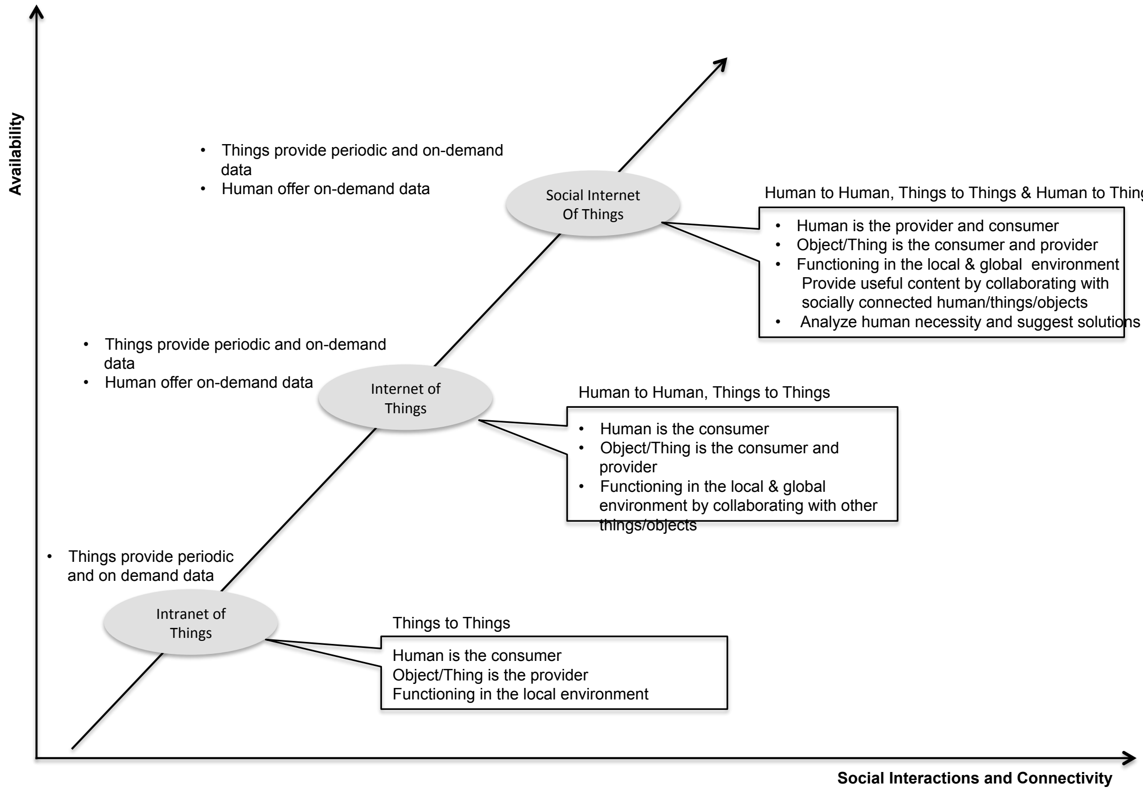

Fig. 3.3 History-line: Evolution graph of SIoT

adaptive layer to selectively combine the strength and remove the weakness based on the applications requirements and data properties in the Chapter 5.

3.2 Social Internet of Things (SIoT)

Social Network Services (SNS) at this point are now being sold as an enormous community of individuals in which the interactions with people in a certain contemporary society are modelled and also talked about. SNSs are made up of nodes of folks, combined with the recommendations including these nodes symbolize the associations of theirs. The overlap between SNS and IoT is lately brought in place inside the literature to allow the relationship of individuals inside the ubiquitous computing universe. Within this particular framework, the data coming from IoT and data coming from SNS opposite human-to-device interactions [20]. In this belief, the worlds on the SNS and IoT are incorporated because of the goal of acquiring social worlds and the physical on the cyber one particular. The ensuing paradigm, viewed as the Social Internet of Things (SIoT), appears to have the chance to permit novel shows together with media treatments for almost any IoT in a profitable as well as cost effective implies [22], though it's essential to find ways for enhancing service intelligence, effective resource visibility, system discovery, object erect evaluation, crowd-sourcing together with assistance system [20, 22].

3.2.1 Evolution of Social Internet of Things (SIoT)

Potential ubiquitous computing will lead to an extensive range of smart services as well as applications to cope with challenges which are numerous that individuals, as well as companies, face in the daily life of theirs via making it possible for crops as well as people getting in contact with whether anyone or anything, in almost any institution, at any second. While IoT analysis [150, 196] have generally reported interaction on the particular bodily society by knowing or maybe actuating from plenty of different items to perform as the biggest novelty, SIoT paradigm, however, raises worries which are essentially about why and how to utilize these services and applications. For this particular objective, you are going to find two considerations as present in Figure 3.3- a) to enhance local community interactions in addition to b) enhancing pervasiveness.

To be capable to settle on every one of the attributes of authentic ubiquitous computing down the road everyday life of ours with increased QoE, we've to improve the connectivity of all of the interactions between humans and things. As mentioned before, for human society, a person typically functions both as a person to ensure that to be a producer to consult with the others; furthermore, in physical or online possibly world, SNs may provide long-range (or maybe perhaps proactive) additionally, on demand (or maybe perhaps reactive) information, e.g., needs, interests, locations, demographic properties, relationship characteristics, etc., of individuals or communities through synergy with each other. Saving, processing, and utilizing these social capabilities may eventually help to increase QoE. Within pervasive environments, owners frequently access expertise anytime and anywhere making use of some kind of things with virtually any sort of interaction networks. When overseen through the SIoT eyesight, pervasiveness will indicate the weaving of actual, virtual, and also real bodily material into SIoT.

Inside SIoT, owners themselves can unintentionally engage in the process of enhancing QoE through products they use as well as discussions on a frequent basis which is short for their interests and needs. Furthermore, variables will collaborate with other many modifications to fulfil the objectives of theirs which are pushed out of the people. In this sense, folks and things aren't seen as specific nodes inside a system, as well as their objectives and needs will weave together building what SIoT is known as by us. That is, these sorts of good link development involving humans and things will result in boosting the accessibility of both parts (i.e., elements and people) and also assuring the transparency of theirs. The two earlier stated notions of transparency and availability might get us with the incredibly pervasive world, as encouraged by succeeding driven ubiquitous computing methods.

3.2.2 Key aspects of Social Internet of Things (SIoT)

To realize a genuine implementation of seamless integration in between social and IoT worlds and to reach the benefits urged by the SIoT vision, several perspectives are deemed. Figure 3.3 illustrates these perspectives coupled with the evolutionary historic past of ubiquitous computing solutions.

1) Interactivity Perspective: The pairing concerning humans and things in IoT can happen in two forms: One) human-to-human or maybe perhaps two) thing-to-thing interaction, as well as definitely it might be achieved using the conventional physical interaction in case of different computer networks and humans giving of problems. In the existing condition of the art, almost all nearly the initiatives focus on one communication type in a time whereas, in this specific thesis, we point out that implementing human-to-thing interactions is essential being the completed viewpoint of SIoT. Such sort of the door is started by interaction to the subsequent level of pervasiveness discovered IoT locations what about the doorstep is opened by point to a large amount of some other networking and communication problems that need to be tackled.

2) Collaboration Perspective: This view appears to be the best essential one in order to learn a whole convergence of the interpersonal plus IoT worlds since human-to-thing interactions are backed by it. We analyze the roles of humans and things. Considering neighbourhood values, SIoT inevitably allows humans and things to function as producers or consumers, and also this results in improving cooperation among all of the entities in addition to unavoidably boosting QoE.

3) Handled Data Perspective: It's also really crucial to consider the type of managing methods in addition info acquisition must be looked at in pervasive environments. We categorize information acquisition methods into 2 categories: one) hands-on particulars acquisition that is crawling methods, learning algorithms, and maybe perhaps several data analysis algorithms along with 2) reactive particulars acquisition which often perform in a real-time way utilizing different data mining and query strategies. Inside SIoT, both types of information acquisition might be used based on the specified scenarios. For example, location info of a person might indicate usually the existing spot for an on-demand query or possibly the historic trajectory by realizing along with analyzing. Moreover, this particular house might well result in the identical state for temperature monitoring by issues.

3.2.3 Socially Connected Objects

Though the basic idea of integration of the sociable aspects with IoT as urged by SIoT is nonetheless brand new or even in a novice stage of searching, investigate contributions nowadays paved precisely how by offering methods for getting people, through Social Network Services (SNS), along with distributed sensors and in addition embedded items as a way to enhance services and applications. Inside [112], for instance, the IoT paradigm is recommended getting enriched with Twitter correspondence capabilities to post info as well as updates about the state of numerous on going tasks and activities. Likewise, within [30], a technique determined by WSN is provided creating also use of Twitter to publish and also share sensory specifics and information. Inside [144], the IoT system is considered a social organization framework to federate ubiquitous IoT building. Various other techniques claim to increase the IoT together with the use of SNs system programming interfaces (APIs), for instance, [29] suggests a wedge allowing individuals to discuss the WWW-enabled products to make sure that others are in a position to use them.

A. Features of SIoT

The SIoT paradigm is owned by an ecosystem that allows actual physical objects plus human to have interaction within an interpersonal framework. Along with this specific framework, applications and services might be given based on Web solutions. To recognize the framework, a few primary building blocks should be supplied. In this specific region, we quantity these important places which form the basis of SIoT: the interpersonal feature, intelligence, socialized devices, and everything as an application.

1) Social Role: In [19, 23, 20], the interpersonal feature initiates from users' SN, the put that the argument to buy it in to IoT group is guaranteeing the device navigability, and also a good system discover. Likewise within [63], the interpersonal functionality is urged by the usage of popular web SNs together with the APIs of theirs to maintain public structure and relationships with smart things. Furthermore, this specific proposed cultural process allows revealing of smart items based on the respect provided by the vicinity. Inside [152], users' SN bookmark profiles are competent to be helpful in program performance for SIoT, for instance, to make use of geolocation information or perhaps post devices' status and updates. The social functionality is discovered in [207] in terms of utilizing SNs for any user interface to control smart objects.

2) Intelligence: In [23], the thought of intelligence is stated as an essential component of the SIoT paradigm which is responsible for creating, updating, as well as terminating the objects' associations in SIoT. This is not simply the assortment of intelligence, within [63], the thought of intelligence is letting effective thing-to-thing service locate basically anywhere smart things are competent to recognize each others' solutions in an instant way. The task offered within [56] envisions that intelligence is used as a middleware merging many remedies as ontologies, techniques for processing user-generated content, and also suggestion strategies. Within a nutshell, intelligence in deep literature appears to be restricted to impartial command methods that run the use of services.

3) Socialized Devices: The concept of socialized goods as launched by very 1st contributions in SIoT as [19, 20, 23] could be the very best essential architectural thing since it indicates the mechanism which many smart objects, as well as embedded solutions, will use to consult with people with the net. Vazquez et al. [182] unveiled the notion of cohesiveness between sensible SNs and objects, an analysis of the features of interpersonal gadgets is available in this specific thesis, focusing on the notion of enabling sensible goods in an effort to chat with some other things, to chat about expertise concerning specific conditions as well as adding to locate support. In [63], social devices rely on Web protocols to consult with owners through an SN environment.

4) Everything as a Service: The concept of changing items along with SNs functionalities into providers and in addition permit them being fast discovered and integrated with various other fixes was furnished in the literature to make use of the convergence between the interpersonal and goods roles as urged by SIoT. So, people are competent to go over the treatments offered by wise things with buddies or perhaps things [93]. This specific sharing style indicates the use of the cultural feature to discover out and also improve services. Nevertheless, the thought of changing each element into an application is supplied in [23] including a wider perspective, by hooking smart products together with the services they offer. The breakthrough of brand new offerings being utilized or perhaps actually mashed up with other providers can happen supported by the interpersonal feature, in that someone can discover trustworthy suppliers within her/his interpersonal world.

Source	Social Role	Intelligence	Social Device	Everything as a Service
[19, 23, 18]			Yes	Yes
[20]	Yes			Yes
[207]		Yes		
[152]		Yes		
[53]	Yes			
[63]		Yes	Yes	Yes
[167, 168]	Yes	Somewhat	Yes	Yes

Table 3.1 Existing Architecture Design of SIoT

B. Architecture of SIoT

In the literature, the web of Things (WoT) is supplied as an evolutionary stage watching the IoT paradigm [63] wherein the former, smart individuals and items rely on Web needs along with protocols to possess interaction as co-workers in an integrated atmosphere. The convergence between the social capabilities in WoT and IoT paradigms formed the focus of huge exploration papers; however, the two terms are now being conversely employed in the literature to connect with almost precisely the same paradigm. In this specific thesis, we use the term SIoT; however, efforts are believed to be by us with the region of SWoT too. Table 3.1 summarizes re search documents with their offered architectural components.

C. Present Status of SIoT

The potentials furnished by SIoT produce obtainable the enhancement associated with a sizeable amount of uses; however, on account on the home member novelty of the idea, it hasn't been completely exploited in a few programs. We split contributions into 2 categories: one) study fashion with the social capabilities in an IoT setting to make a better quality of system together with 2) a few of fully or partially implemented prototypes that suggest the notion of like the SNs with IoT to correct puts the daily life of ours.

Different analysis fashion takes place in the literature, though they're not initially provided underneath the SIoT umbrella, they are experienced to still press into it by merging the interpersonal element with IoT. We quantity up every among all the with regards to Table 3.2. The evaluation trends usually include semantic web plan locations to allow community service to discover and also devices and mash up. Different method concentrates on understanding numerous interpersonal abilities from internet SNs or

Source	Semantic Web Service	Social Cognition	Location based Service	Social Graph Analysis	Trust Management
[23]	Yes				
[152]	Yes				
[63]	Yes				
[139]				Yes	
[36]				Yes	
[93]	Yes				Yes
[120]		Yes	Yes		
[134]		Yes	Yes		
[12]		Yes	Yes		
[145]					Yes
[28]					Yes
[167, 168]	Yes	Yes	Application based	Yes	Yes

Table 3.2 Research Trends of SIoT

perhaps probable peer-to-peer SNs to provide favourable careful services. Location-based recognition remains an extra study design by which region data are collected as well as magnified offering personalized services. The evaluation of the SN graph constantly is additionally deemed to increase knowledge of the interpersonal interactions. Lastly, loyalty management is provided to orchestrate the procedure of method discovery. Table exhibits a selection of the prototype circumstances in terms of SIoT.

Product	**Industry**	**Social Features**	**Connectivity**	**Open API**
Pepsico: Social Vending System[11]	Beverages	Socially connected vending machines Gift/share a beverage with a friend with video messages	Wired	No
Nike: Nike+ FuelBand[12]	Apparel, Accessories	Fitness-tracking wristband social network. FuelBand users sharetheir fitness data	Wireless	Yes
Corventis: Nuvant MCT[13]	Healthcare	Noninvasive, ambulatory arrhythmia monitoring physicians, patients, and families coordinate	Wireless	Yes
Good Night: Lamp [14]	Home Appliances	Share your presence and availability in a circumambient way	Wired	Yes

Table 3.3 Commercial products similar to SIoT concepts

Source	Gastronomy	Smart Shopping	Smart City	Smart Home
[56]	Yes			
[46]		Yes		
[184]			Yes	
[102]				Yes

Table 3.4 Available domains of SIoT Prototype

Several commercial sectors have approached the current phenomena by object itself employing leading edge technologies to produce items which are brand new also as services that will own the consequent trend of improvement in SIoT. These public gadgets that vary from low-cost sensors to highly effective inserted strategies are in a position to collect info and also speak this info together with the net to SNs of people and devices who might respond to a problem, provide a service, and increase a solution [166]. Table 3.4 summarizes a choice of off-the-shelf manufacturing sociable models. These are truly appealing products that use the completely new knowledge of SIoT on the customers. While except things from that offer uncovered APIs for third party providers to possess interaction with the equipment of theirs, almost all nearly the products accessible on the industry keep on to get not good at mingling with some other third party gear along with software, this is a battle simply for the forthcoming interoperability of SIoT strategies.

In order to standardize the SIoT features and activities, few standardization organizations accept initiatives associated with the issue of SN attributes towards the IoT, as well as unquestionably the includes IoT objects with SNs principles.

For example, ITU-T [15] focuses information and communication technologies. Proposal in [118] combines SNs and IoT are part of the standardization activities oneM2M [16] started in 2012. AIOTI's [17] key aim is to enhance innovation and economic development across the Internet of Things in Europe. To maintain European competitiveness on a global scale, this aim is best achieved by working not just within Europe, but also with key enabling actors across the world. Thinking about the benefits of SNs within the framework of IoT as well as in M2M, it is anticipated that a there will be some similarities with SIoT.

Todays ubiquitious computing introduces a wide scope of smart services and applications to adapt to numerous difficulties that people, society and organization face in regular activities by means enabling people and things to be associated with it is possible anyone

[15] http://www.itu.int/
[16] http://www.onem2m.org/
[17] https://aioti.eu/collaborations/

or anything, in any place, at any time. While IoT contemplates have normally referenced communication to the physical world by detecting or impelling through a considerable lot of various devices to be the greatest innovation, SIoT vision, in any case, raises significant worries concerning why and how to use these services and applications. For this goal, there are two contemplations as appeared in Figure 3.3 expanding social interactions and improving availability. To choose every one of the properties of genuine omnipresent figuring in our future everyday life with high QoE, we have to improve the availability of a considerable number of connections among humans and things. In human culture, an individual normally capacities both as a consumer and as a producer to interact with others; besides, in either physical or online world, SNs could give long haul (or proactive) and on-request (or reactive) data, e.g., needs, interests, statistic properties, relationship qualities, and so forth., of people or networks through a joint effort with one another.

3.3 Summary and Discussion

In this chapter, we have presented the concepts and technologies we have considered in this thesis. IoT is the main area we have focused in this research. We have surveyed the present status, past story and future direction of IoT including the enabling technologies. We have also discussed the Semantic web and related technologies and their relation and with IoT. We have also surveyed the current status of the comparatively new research area Social IoT, which is basically a integration of Social Network and IoT. Another related technology called stream data processing has been discussed as well. Stream data processing is adopted in this thesis to deal with the continuous data from SIoT objects.

Chapter 4

Social-IoT Management Framework

In this chapter, we present the SIoT Management Framework(SIoT-MF). It manages and monitors SIoT objects intelligently and cognitively according to the context in real-time. SIoT-MF is an integration of the techniques and technologies studied and designed in this thesis. SIoT adopted IoT and social network platform to take the benefit of connected objects and socially-connected human respectively. To attain the best out of "objects social network", we need to enable social relationships among objects. We also integrate virtulization in the SIoT-MF to combine the functionalities of the SIoT objects. Moreover the integration of real-time processign enable the real-time monitoring of the objects. We consider the following four types of objects in our proposed SIoT-MF i.e. a) *Real-world Physical Object (PO)*; b) *Virtual representation of PO called Virtual Object (VO)*; c) *Human skills as an object called Abstract Object (AO)* and d) *Combination of different types of VO and AO called Composite Virtual Object (CVO)*.

SIOT-MF ensures semantic relationship among VOs and composes new services by combining VOs that we call CVO. Additionally, we identify the human knowledge, expertise and services as the Abstract Object (AO). The reason behind identifying AO is to enable visibility and accessibility of human capabilities through a cognitive framework in such a manner that society members can share their capabilities with each other as a contribution to the society. SIoT-MF is capable of combining any number of VOs and AOs to form a new CVO. This ability of SIoT-MF makes it unique among existing Social IoT approaches [22, 78]. People may offer their capabilities and/or share their resources either for free of charge or with a fee. We also enable real-time interaction by using stream processing techniques in the SIoT-MF to monitor the SIoT object in real-time.

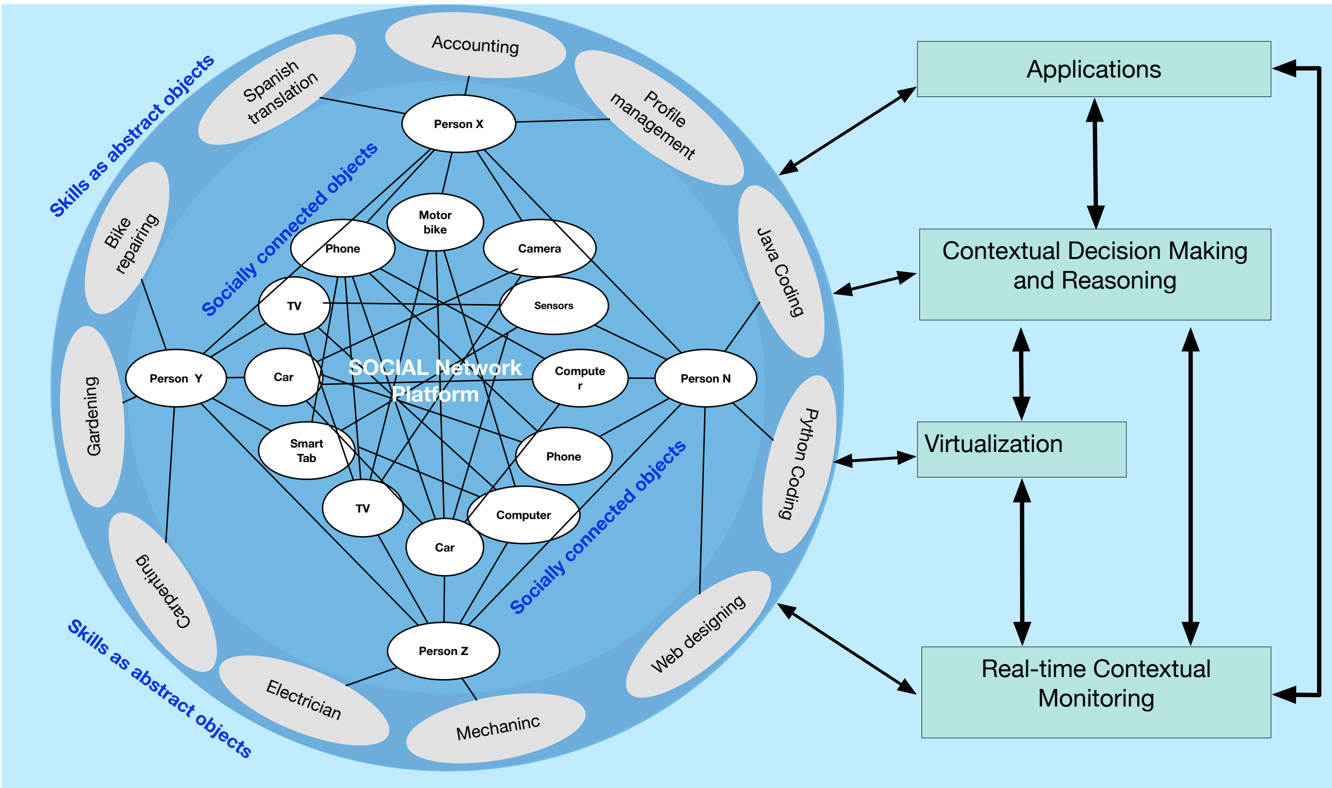

Fig. 4.1 Topological view of the SIoT-MF

We organize this chapter in the following sections, the functional design of SIoT-MF and key requirements are elaborated in the Section 4.1. The SIoT-MF architecture and the interactions among all the components are presented in the Section 4.2. We describe a real life scenario to show the process of SIoT-MF in the Section 4.3 and finally we summarised this chapter in the Section 4.7.

4.1 Key Functionality Design of SIoT-MF

SIoT-MF is designed to enable the cognitive-contributory societies where human and objects can collaborate with each other to create a service on demand. The topology in the Fig 4.1 shows the requirements of different modules and techniques for planning, designing and modelling of the architecture of SIoT-MF.

We considered several recent technologies including IoT and Social Network are considered to achieve desired functionalities of SIoT-MF. We have detailed all the related

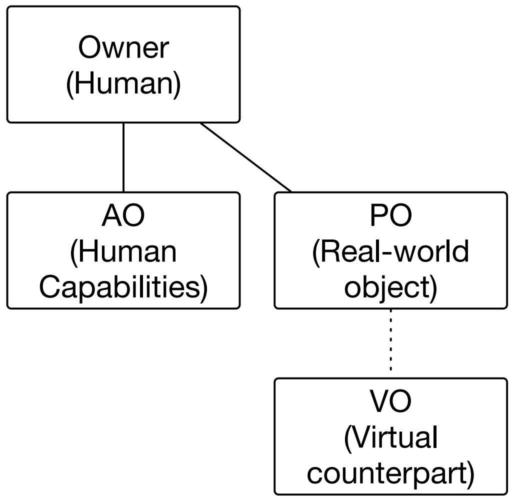

Fig. 4.2 Taxonomy of SIoT object

technologies in the Chapter 2. Here, we define the key functionalities we consider to address the requirements.

4.1.1 Physical object profiling and object virtualization

As an initial step, every SIoT object has to register its profile at the SIoT-MF. According to the best of our knowledge there is no existing social network platform in practice that accepts physical object other than human. Apart from the human social network, we assume a social network platform that accepts physical objects profile to be registered either by the owner or by the smart object itself after a authorization from its owner. As a result SIoT-MF creates a corresponding VO for each PO and stores in the registry. Human capabilities are need to be attached with an owner's profile as an AO so that SIoT-MF stores this information to the registry and uses it according to the user requirements.

POs are virtualized to simplify the object collaboration among VO and AO for creating a service. This mechanism also enables the objects to be identified and actuated through their own profile that has been created during the registration process in social network. Moreover, VO and AO maintain a semantic relationship with each other. The virtualization is adopted in the SIoT-MF. Additionally, we have extended the object virtualization method, techniques and policy in the to be compatible with SIoT objects.

4.1.2 Semantic relationship and interoperability among SIoT objects

Semantic interoperability explains the ability of different segments to access, infer and interpret data and knowledge in a certain domain. SIoT objects need to exchange data with other objects and with human as well. The key enabler here is to defining, annotating meaningful description of the objects and its data for gaining the interoperability. In the SIoT-MF, two ontologies are proposed i.e. *SIoT ontology- objects are annotated semantically according to this ontology* and *QoS ontology- the QoS criteria of the objects are annotated according to this ontology* to ensure the semantic relationship, interoperability and reusability. These two ontologies are connected through the common VO node in the both ontologies.

4.1.3 Constraint and QoS aware VO selection and CVO composition

The efficiency of the SIoT-FM depends on selecting a suitable candidate object from a pool of capable candidate objects for executing a given task. SIoT-MF analyses the user requests and select a suitable VO from the available VO registry according to the user constraints and objects QoS criteria.

Our investigation for a selection mechanism included selection mechanisms in economics as well. Thankfully, we found a Simple Additive Weight (SAW) based selection mechanism called Multiple Criteria Decision Making(MCDM) in economics. However, this mechanism lacks the ability to consider objects' constraints during the selection process. In SIoT-FM, we use MCDM when a task needs only one VO, as a result, no CVO is created.

We consider another selection mechanism i.e. Integer Programming (IP) where VO constraints are considered in the calculation to avoid dependency among VO in a CVO. In SIoT-FM we use IP when more than one VO is required in a CVO for avoiding dependency among VO. In the next chapter, we will describe the entire selection process in detail.

4.1.4 Real-time monitoring

In the SIoT as well as IoT paradigm data are generated continuously and user applications need the data processed continuously either real-time or at least close to real-time for preserving the data freshness, it means more new data are more relevant. Hence, another important aspect of our proposed SIoT framework is to process data in real-time. Traditional data processing techniques can process static data but not suitable for ensuring data freshness particularly when data and application requirements are a continuous process in real-time. To cop-up with this situation we consider stream-processing technique in the SIoT framework. SIoT objects operate in distributed and dynamic environment and to ensure their freshness we use real-time monitoring using the stream processing engines.

4.1.5 Categories of object relationship

We assume that all physical and abstract objects are owned by human and the objects can join different communities with the permission of their owner. Apart from the owner's relationships- objects are also able to initiate a different kind of relationships with each other [22]. We define relationship categories below,

Parental or owner object relationship This type of relationship connects the same manufacturer's same model of objects e.g., cars, mobile, electronic devices etc. This relationship sustains until the lifetime of a product in the market given by the manufacturer.

Geographical relationship Established among homogeneous and heterogeneous objects within a same geographic location(assigned by the owner) e.g. an owner may allow his/her refrigerator object to connect all other refrigerator objects within same geographic area even if the owner is not directly or indirectly connected with all other owners.

Co-work object relationship An objects may initiate a relationship with other member objects when a number of objects combinedly participate to serve a common interest.

Inherited relationship A new object may inherit the relationship from an old object by registering to the platform using the same unique identifier. However, parental object relationship could be different based on the manufacturer and model.

Social object relationship Objects try to establish a relationship when they come into contact frequently, e.g. workplace, classroom, church etc.

Ad-hoc relationship For an ad-hoc task an object may try to initiate a relationship among surrounding objects for a short amount of time depending on the requirements. The object may initiate a co-work relationship consequently.

It is worth mentioning here that, human intervention will not be required when an object initiates a relationship if the owner sets the priority and settings beforehand. Otherwise, by default objects ask for permission and authentication to ensure security and privacy.

4.1.6 Context awareness in SIoT-FM

Contextual decision making is necessary in smart environments. In the SIoT-FM, contextual decisions are made based on the data collected. e.g. In the SIoT enabled smart home a refrigerator (SIoT object) is able to send a notification to the owner before she/he runs out of an item. It also recommends a product by collecting review from other socially connected refrigerators and set a reminder to buy it from a shop.

4.2 SIoT-MF Architecture

This section presents the architecture of the proposed SIoT-MF to address the research questions defined in Chapter 1. The functional components are shown in Figure 4.7 through the five functional tiers. We discuss each of these tiers in details as below;

4.2.1 Tier-1: Real-world physical object and abstract object

This tier bridges the gap between real-world and virtual-world. The real-world physical object and abstract object tier include devices, sensors, human, human skills and expertise and any other IoT object. This tier interacts with the real-world and fetches data continuously to send them to the Tier-2. Social network platform handles objects' profiles and other related information. A PO-profile includes its functionalities, capabilities, brands, model, type, servicing date, warranty, specifications etc. On the other hand, an AO profile includes human capabilities. PO and AO both are owned by human profiles. The social network profiles work as an interface to connect to the virtual world. Moreover, AOs are also considered as an object and connected with human POs. In all cases, PO's and AOs are allowed to communicate with other objects only if owners' privacy policies allow

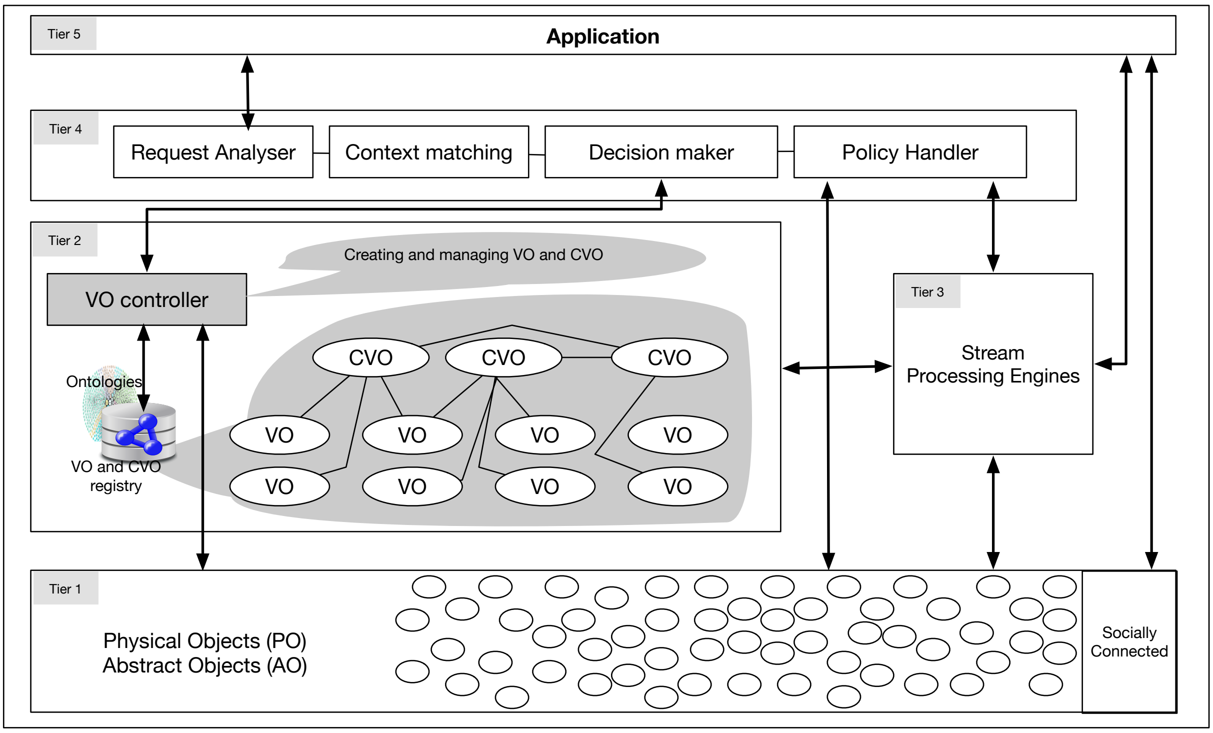

Fig. 4.3 Functional components architecture of SIoT-MF

to communicate. Objects may register to the social network platform directly and also communicate with other objects freely if the owner authorizes the object beforehand.

4.2.2 Tier-2: Object virtualization

The limitations in the processing capability of sensors and actuators are a great challenge for the entire IoT paradigm. Things even get worse when it comes to the real-time applications where continuous processing of data is the key to preserve information freshness. In such circumstances, a convenient setting is required to enable those objects and their functionalities to the cyber world.

Virtualized object known as VO has become an important part in many IoT platform to handle the resource and flexibility issues that are quite difficult or sometimes impossible at the real-world object level. VOs are harmonized and composed to form application level features to satisfy the service requirements. This tier is responsible to create a virtual representation of physical objects and abstract objects.

In the SIoT-MF, VOs are created automatically whenever a PO registers to the SN platform. VOs have

```
Date and Time of Creation:
Identification number:
GPS Location:
Owner Profile url:
Type of the object:
Model of the object:
Version:
Manufacturer name:
```

Listing 1 VO profile template

```
<http://xmlns.com/foaf/user1> <http://SIoT.insight-centre.org/owns> <http://SIoT.insight-
    centre.org/PO/refrigerator/PO-ID>
<http://SIoT.insight-centre.org/PO/refrigerator/PO-ID> <http://SIoT.insight-centre.org/
    hasLocation> <pimo: http://www.semanticdesktop.org/ontologies/2007/11/01/pimo#lt53
    .2899407-lo-9.07426380000004>
 <http://SIoT.insight-centre.org/PO/refrigerator/PO-ID> <SIoT:http://SIoT.insight-centre.org/
     isRepresentedAs> <http://SIoT.insight-centre.org/refrigerator/VO-ID>
<http://SIoT.insight-centre.org/refrigerator/VO-ID> <http://SIoT.insight-centre.org/capability>
     <http://SIoT.insight-centre.org/capability/buyOnline>
<http://SIoT.insight-centre.org/refrigerator/VO-ID> <http://SIoT.insight-centre.org/capability>
     <http://SIoT.insight-centre.org/checkBarcode>
<http://SIoT.insight-centre.org/refrigerator/VO-ID> <http://SIoT.insight-centre.org/capability>
     <http://SIoT.insight-centre.org/countItem>
<http://SIoT.insight-centre.org/refrigerator/VO-ID> <http://SIoT.insight-centre.org/capability>
     <http://SIoT.insight-centre.org/checkExpiry>
<http://SIoT.insight-centre.org/refrigerator/VO-ID> <http://SIoT.insight-centre.org/capability>
     <http://SIoT.insight-centre.org/socialNetworking>
<http://SIoT.insight-centre.org/refrigerator/VO-ID> <http://SIoT.insight-centre.org/isAvailable
    > "date-time"
<http://www-sop.inria.fr/edelweiss/fabien/docs/w3c/rdfsource/rdfsource.html> dc:creator <http
    ://ns.inria.fr/fabien.gandon/foaf#me>
<http://ns.inria.fr/fabien.gandon/foaf#me> rdf:type foaf:Person
<http://ns.inria.fr/fabien.gandon/foaf#me> foaf:name "Fabien Gandon"
<http://ns.inria.fr/fabien.gandon/foaf#me> foaf:mbox <mailto:fgandon@inria.fr>f:mbox rdf:
    resource="mailto:fgandon@inria.fr"/>
```

Listing 2 Sample dataset

extended the functionality of POs by organizing them in a semantic ontology model in the virtualization tier. We provide a sample VO profile template and a sample VO dataset in the Listing 4.2.2 and Listing 4.2.2 respectively. As a result, the POs are connected socially in the real-world tier and the VOs are connected ontologically at the virtualization tier. The Composite Virtual Objects (CVO) are formed by combining the functionalities of VOs as shown in the Figure 4.4. Intelligence is required at the composition level to form a CVO (to meet dynamic requirements even at the real-time). Another important feature is reusability of a CVO. Rarely used CVOs are removed from the index when the systems are resource constrained. Virtualization tier maintains a registry to log the active CVOs and

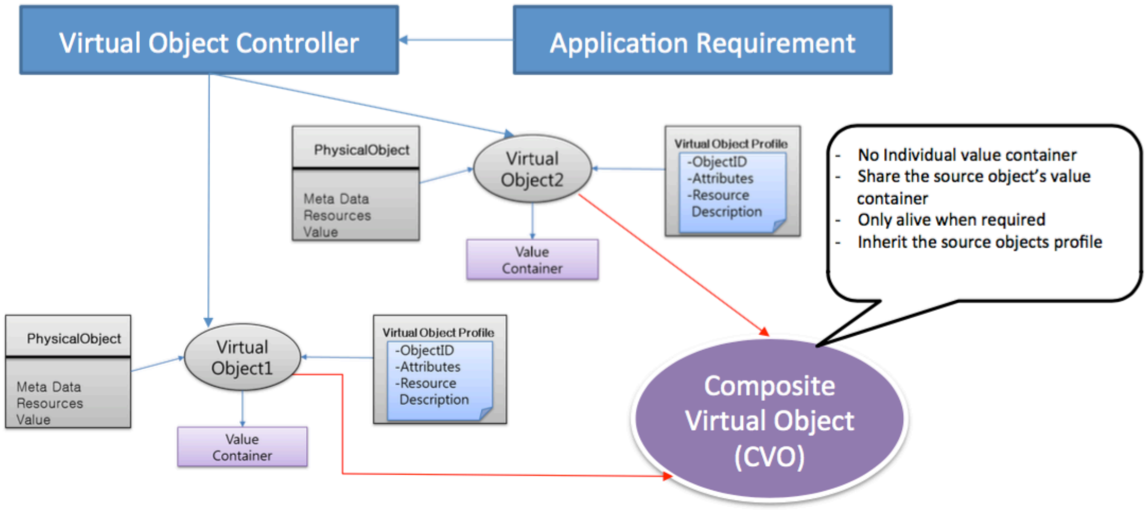

Fig. 4.4 VO and CVO mutation

the resource consumption of all these CVOs are proportional to a threshold of available resources of the local systems. When the requirements are identified, the system needs to check how it can compose the required service by reusing indexed CVOs at a maximum level.

The first step is to search the CVO index to find a potential match, an unsuccessful index search refers to the second step to form a new CVO when there is no match in the registry. The registry contains,

- CVO identifier with a date
- List of sorresponding VOs that are involved in the CVO
- Corresponding POs that are source of all VOs
- Privacy settings of the PO's profile.

This procedure also enables the devices to be identified and actuated through their own device profile, method and control information. In fact, all virtual objects maintain a semantic relationship with each other which has been described in the next section.

A virtual object profile template is provided in the Listing 4.2.2

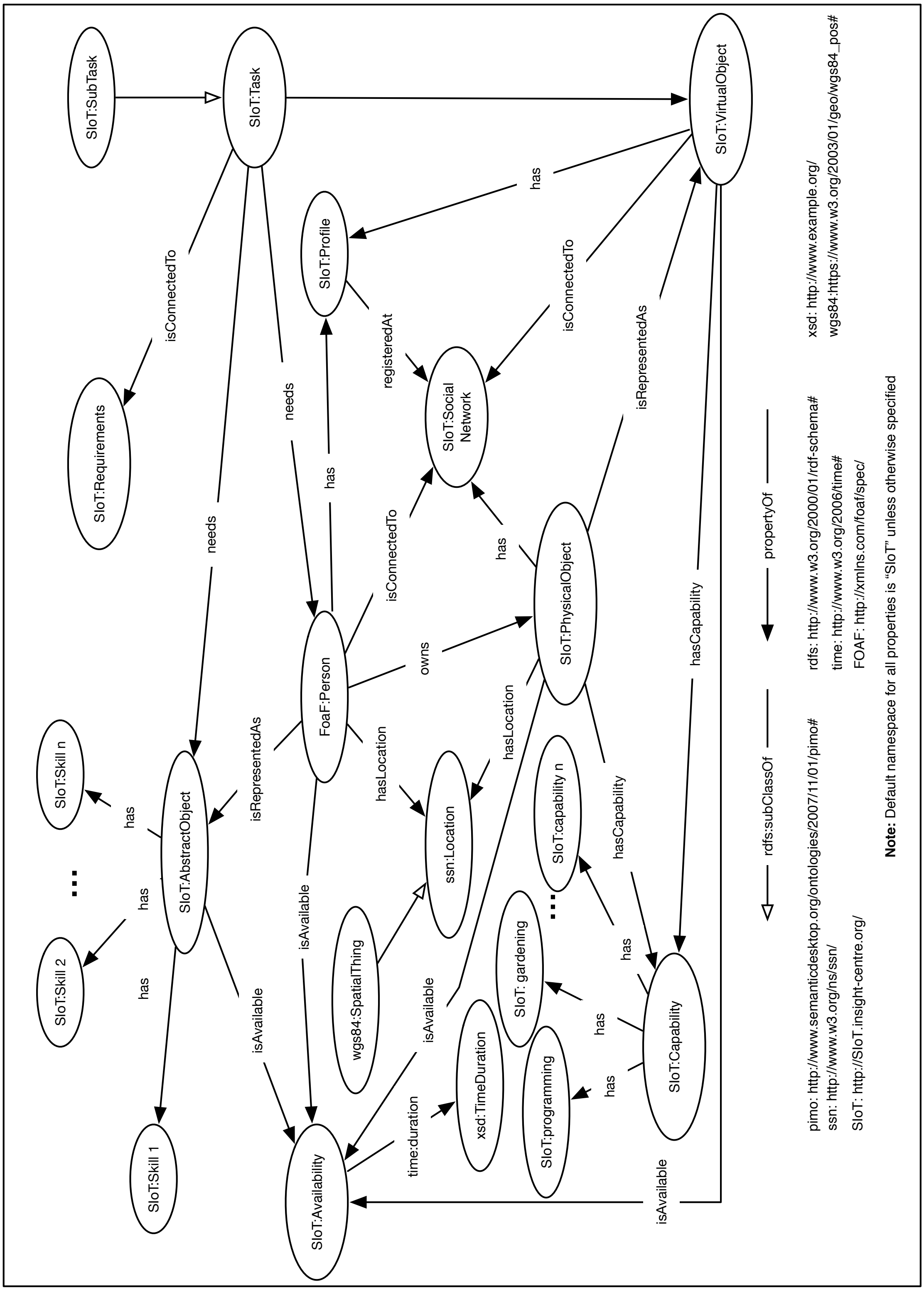

Fig. 4.5 SIoT ontology model

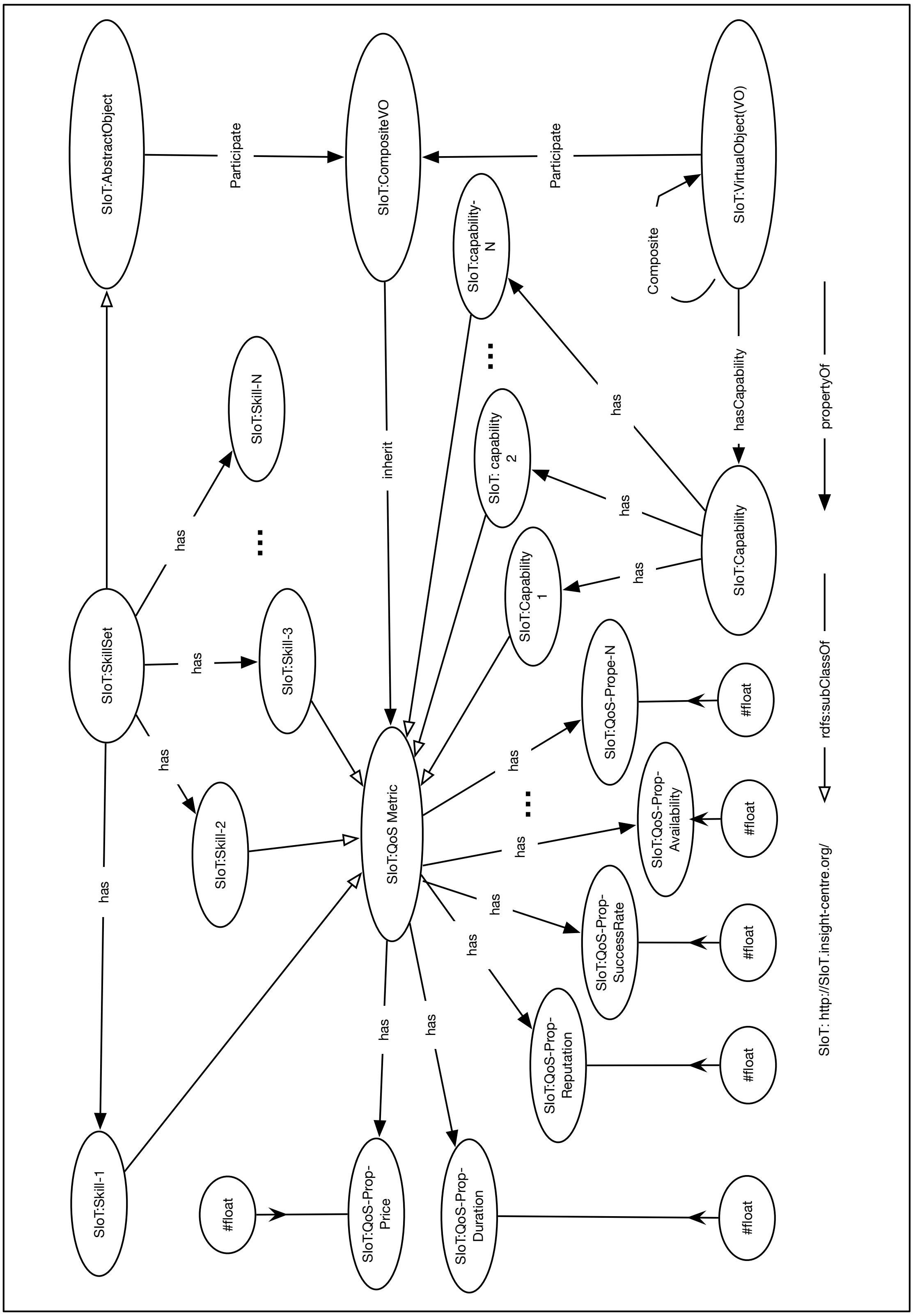

Fig. 4.6 QoS ontology of SIoT objects

Semantic Ontology Model

We used several existing ontologies for representing the relationship among VOs and AOs Figure 4.5 depicts the ontology model of VO and AO for their relation to satisfy requirements.

We used semantic representation techniques and have used few existing semantic ontology models and proposed our model to represent different objects (virtual object, physical object and abstract object). We also represent users' tasks and their requirements. Our semantic ontology model also captures the interaction among task entities and their performance regarding QoS. Fig 4.5 showcases a complete picture of our information model for SIoT Framework. Below we briefly explain a few major concepts in SIoT ontology;

FoaF:Person, we used the concept of a person represented in Friend-of-a-Friend(FoaF) Ontology, which is a well-known ontology to represent persons, their contact details and their friends list. *SIoT:PhysicalObject* represents any physical object owned by a *FoaF:Person*. Every *SIoT:PhysicalObject* is also represented virtually as *SIoT:VirtualObject*. Both *FoaF:Person* and *SIoT:VirtualObject* have their corresponding *SIoT:Profile*, which is registered over an *SIoT:SocialNetwork*. Every *SIoT:VirtualObject* has a set of capabilities (*SIoT:Capability*) which can be performed by *SIoT:PhysicalObject*, we also listed capabilities for its virtual representation *SIoT:VirtualObject*.

SIoT:AbstractObject is an abstract representation of a *SIoT:Person*, similar to *SIoT:VirtualObject*, which is a virtual representation of *SIoT:PhysicalObject*. Every *SIoT:AbstractObject* contains a set of skills, which are represented as *SIoT:Skill*. Skills attached to abstract objects represent skills of a human associated with the abstract object.

SIoT:Availability denotes whether an abstract, physical and virtual object is available to perform a specific task based on its capabilities. *SIoT:Availability* has also a time duration associated with it, which indicates that whether the selected object is available to perform a task during that particular time span. *SIoT:Task* indicates a set of requirements to perform a given task at hand, every task can contain a set of *SIoT:SubTasks* for its completion. *SIoT:Task* can be performed by any available abstract, physical and/or virtual object as well as a person who has a set of desired capabilities or skills required to perform a task.

Figure 4.6 presents the QoS ontology model of SIoT objects. This ontology model depicts the relation among QoS criteria and SIoT objects.

SIoT:SkillSet is a subclass of *FoaF:AbstractObject* which has a set of human skills. These skills can be performed by human whic is shown in Figure 4.5. *SIoT:SkillSet* has several skills and each skill has QoS five criteria (we considered five criteria in this thesis). All five QoS criteria are *SIoT:QoS-Prop-Price*, *SIoT:QoS-Prop-Duration*, *SIoT:QoS-Prop-Reputation*, *SIoT:QoS-Prop-SuccessRate* and *SIoT:QoS-Prop-Availability* will have ranking score from the user after each completed task. *SIoT:QoS-Prope-N* shows that a new criteria can be added in the ontology. *SIoT:QoS metric* is the metric of all available QoS criteria. *SIoT:Capability* is the set of objects capability whis is performed by the object itself. Every *SIoT:Capability* i.e., *SIoT:Capability 1* to *SIoT:Capability N* also has five QoS criterai like *SIoT:Skill.* the difference is human has skills and object has capalities.

4.2.3 Tier-3: Real-time stream processing

```
REGISTER QUERY Monitoring AS
Prefix owner1: <http://mysmarthome.com/>

SELECT ?object-1 ?object-2 ?object-3 ?object-10
From STREAM <http://mysmarthome.com/object1> [RANGE 10m STEP 10m]
From STREAM <http://mysmarthome.com/object2> [RANGE 12h STEP 10m]
From STREAM <http://mysmarthome.com/object3> [RANGE 24h STEP 10m]
From STREAM <http://mysmarthome.com/object10> [RANGE 7d STEP 10m]

WHERE { ?object-1 ?object-2 ?object-3 ?object-10 .}
```

Listing 3 Query (CSPARQL) for monitoring objects.

For SIoT, we consider two types of streaming data, firstly a streaming data where any virtual object can join on the fly describing its capabilities or skills, which will be added into virtual objects registry indicating their availability to perform different tasks. Secondly, we consider stream data produced by users while submitting tasks and their requirements. In order to process these data streams, we have considered two types of stream processing engines. RDF stream processing engines which are capable of processing RDF data (users tasks processing and matchmaking) and to enable non-rdf data we considered Apache SPARK[1].

This tier (stream processing) is connected with *Tier 1* (PO and SN platform) *Tier 2* (virtualization), *Tier 4* (Decision making and reasoning) and if required this tier can also provide output directly to the application. Tier 4 registers queries according to application requirement and receives the output stream from stream processing tier, VOs and CVOs get the profile updates from this tier including other information if required. A sample

[1]https://spark.apache.org

query is presented in the Listing 4.2.3 The clients are authenticated through social network platform and utilize an interface for sending the requests to SIoT-MF. Decision maker translates the application request to semantic VOs and CVOs. This mechanism represents the reasoning for the automatic user request translation into a machine-understandable format. VOs and CVOs are mapped and/or created according to the requirements received from the decision maker.

The clients are authenticated through Social Network platform and utilize proper interfaces for sending a request to SIoT-MF. Decision maker translates the application request to semantic VOs and CVOs. This mechanism represents the reasoning for the automatic user request translation into a machine-understandable format. VOs and CVOs are mapped and/or created according to the requirements received from the decision maker.

4.2.4 Tier-4: Contextual Decision Maker

In the SIoT-MF, the *decision maker* considers available objects and selects the best suitable objects for executing a task. Hence, analysing and scaling is the first step to create new CVO. It also considers the corresponding SN account holder and their objects security policy and restrictions before selecting or forming a new CVO. Real-time stream processing component continuously plays a very important role in processing and querying the desired information to the decision maker. Selecting a suitable VO and AO is a complex calculation because of their different types of capabilities and dynamicity. We use two different techniques Multi-Criteria Decision Making (MCDM) and Integer Programming (IP) to select a suitable object. In the next chapter, we detail these two techniques.

4.2.5 Tier-5: User Application

User applications are the main component of this tier. Users are expected to send their request through an user interface along with their preferences on QoS criteria of objects.

4.3 Process and Scenario

This section presents a real-life use case of SIoT-MF. We assume that SIoT enabled smart appliances are available where human and physical objects are socially connected through

SIoT-MF. We use the SIoT-MF concept as the core system for showcasing the practicality of SIoT-MF.

Different smart applications are being developed recently to make human life smarter and easier. These applications collect data from various sources, e.g. IoT (Internet of Things) devices, Social Networks, e-commerce sites, Smart Cities, etc., and analyse them to retrieve meaningful insight from random data sources [1-3]. Many people prefer online shopping to meet their needs in everyday life since it minimises the hassles of going to malls and buying items in person. However, they have to perform the tedious task of searching the online to buy the items. On the other hand, the other group of people, who prefer shopping 'On-site', frequently struggle to manage time due to the necessity of searching for different items in different shopping malls. Again, many people usually find it cumbersome to maintain an organised check-list of all the items running out. Thus they at times forget to buy some items and/or buy many unnecessary items due to the suspicion of the shortage. We believe this hassle can be minimised if an assistant helps people to maintain the list of required items, and suggests them the stores from where they can buy good items easily with low cost. It is very much possible today to gather the required data to make the necessary analysis for such decision making, thanks to the IoT-based sensors in home appliances, online social media and different recommender systems. Hence, we intend to present a smart SIoT-MF based shopping system to support the users to enjoy smart shopping experiences with the help of an efficient shopping guide. To leverage this trend, we propose a smart shopping system by using SIoT-MF.

4.4 Motivation

The SIoT-MF enabled use case is motivated by following reasons,

Minimizing fake recommendations The rise of online marketing and review systems introduces flexibility for the consumers to evaluate before buying a item. Nevertheless, it also introduces a number of fake recommendations and paid reviews of different stores, which is very disappointing for the consumers because of wasting money, effort and time [4-10]. Minimizing the dilemma is not trivial. In this use case, a smart refrigerator rely on the social network to collect recommendations from connected friends, trusted groups and community where other friends provide reviews and offers of the items. Most of the time social network friends honestly express their opinions while recommending a item to a friend, when it is not influenced or promoted by any item marketing service. We believe that this approach is helpful for both consumers and shop owners, where consumers can get rid of false reviews and shop owners can achieve customer satisfaction, provided that most of the online and on-site shop owners sell item from different sellers and they also need to control the fake review problem for the sake of their reputation.

Saving time and transportation cost In the modern era, many people are very busy or too distracted to ensure their sufficient food stock in the house. The main challenge for them is managing time for go to shop frequently. To save time sometimes people take home delivery which may involve extra transportation cost for each individual item. We have addressed this time and transportation problems in the use case and provides a reasonable method deal with the problem. The socially connected refrigerators make the list of necessary items and communicate with friends and neighbours who also need to go for shopping at nearly same time. The SIoT based shopping systems are able to save time and transportation cost and can be handled by the socially connected smart refrigerators.

Notification based on time and location The smart shopping application includes reminder system to notify people for buying necessary item on time. Moreover, it also notifies people when they pass by a designated shop from where he suppose to buy some items. This location based reminder system is helpful for the people who prefer on-site shopping.

4.5 SIoT-MF Enabled Smart Shopping System

A SIoT-MF enabled smart shopping system includes socially connected smart refrigerators and its owner. A smart refrigerator generates requirements to buy new items following the status of different items monitored by the sensors [46-52] in the refrigerator. SIoT-MF selects the most suitable item based on the QoS criteria and user preferences. We have described the selection process in the Chapter 4. Eventually, the other modules in different parts of the SIoT-MF are triggered.

Social Network The social network segment in the Figure 4.7 shows the presence of a SIoT object and their friends. A smart refrigerator gets recommendations for various items from other connected friends. Many recommender systems rely on the ratings/suggestions made by unfiltered users. However, nowadays it is not easy to select a item based

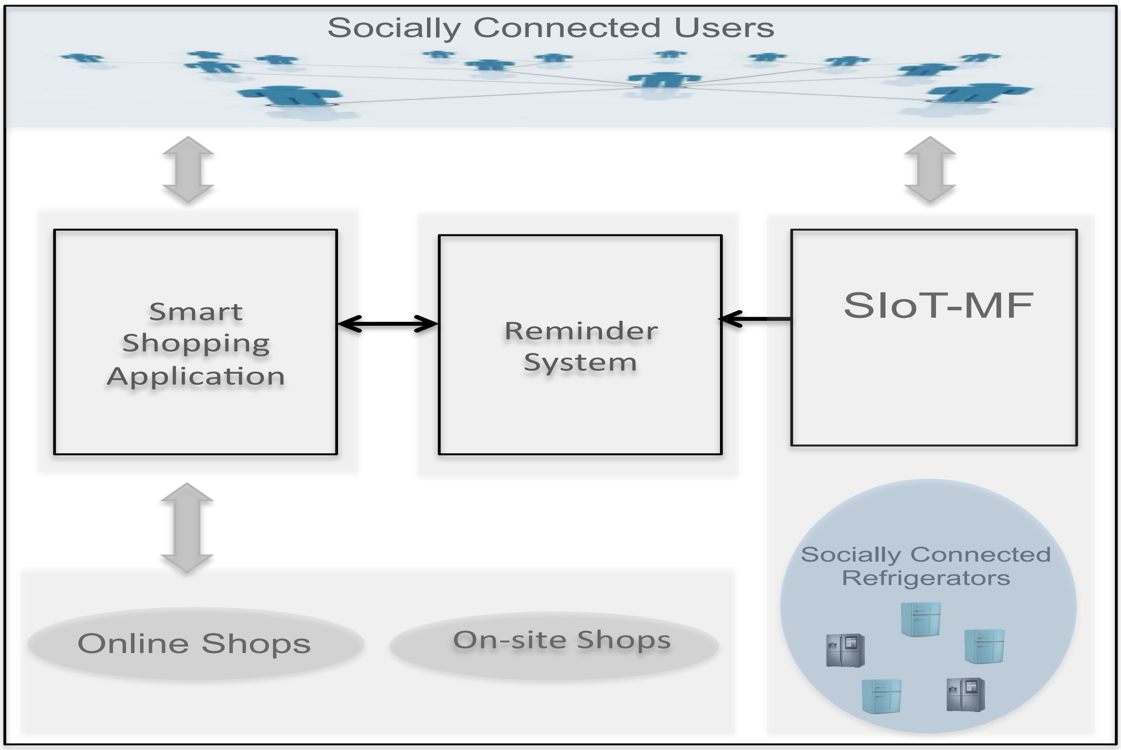

Fig. 4.7 SIoT based smart shopping systems

on all such online reviews because of the possibility of fake reviews when the reviewer is unknown. In contrast, we rely on social network friends, family and known person for advice, suggestion and recommendation. For this reason, our proposed system makes recommendations for different items based on the reviews and recommendations done by the people in the friend-list of a user.

Smart Shopping Application This segment is an application, using which a user can access the functionalities of the system. It stores the selected store locations after getting the suggestions from the recommender system and the reminder system. It also has a module to work as an assistant to execute shopping online.

Reminder System The reminder system collects shopping list from a smart refrigerator and sets notification alerts for users for buying item. The system is two folded, it reminds user based on a scheduled time and/or uses GPS location to remind a user when s/he passes by a designated store to buy the selected item.

Online Shops This segment consists of online shops API to shop online, and to collect data of available items along with new offers in different stores. The smart application is responsible for shopping online with the user's consent.

On-site Shops The shopping application We assume that On-site shops are IoT-enabled with shopping ontology, and stores/shopping malls have the facility to allow a user to connect to the shopping systems. When an user arrives at a shop, the smart shopping application guides the user exactly to the items, so s/he does not need to spend time unnecessarily. This feature saves time because the user know exactly where the item is. Additionally, it also minimizes unnecessary shopping as users do not walk around for searching an item and buy an additional item.

4.6 Smart Shopping Use Case using SIoT-MF

In this use case, we consider SIoT-enabled smart refrigerator as a SIoT object which is capable of measuring the items quantity, type, expiry date, shop name and company name. A smart refrigerator (SIoT node) analyse the data of each item and get recommendation from socially connected friends and sends a selected item and shop name to the shopping application. Figure 4.9 shows a generic flowchart of the smart shopping systems called "Grocery Offer and Review in Galway" from where it is able to collect review and special-offer of various grocery items.

Alice's refrigerator detects that he needs to buy milk otherwise he will be out of milk in 3 days. At this point, the refrigerator pass this information to SIoT-MF including its preferences on QoS criteria i.e. Price, Reputation, Transportation Cost, Duration. SIoT-MF collects the value of each criteria from Alices's refrigerator and his social network friends. And then select the most suitable item by you applying the technique described in the Chapter 4. We provide an example to buy Milk following using SIoT-MF based shopping application.

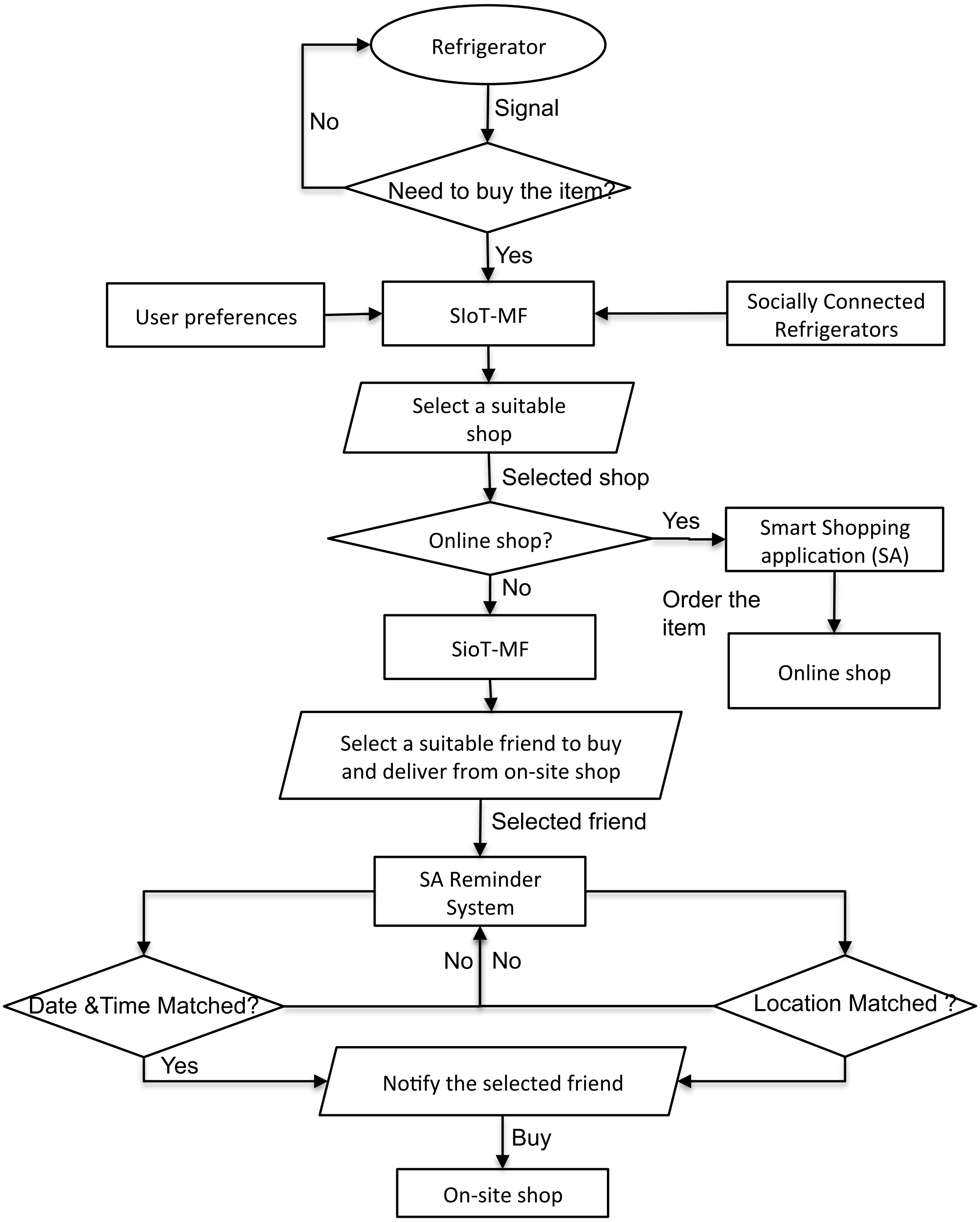

Fig. 4.8 SIoT based smart shopping flow diagram

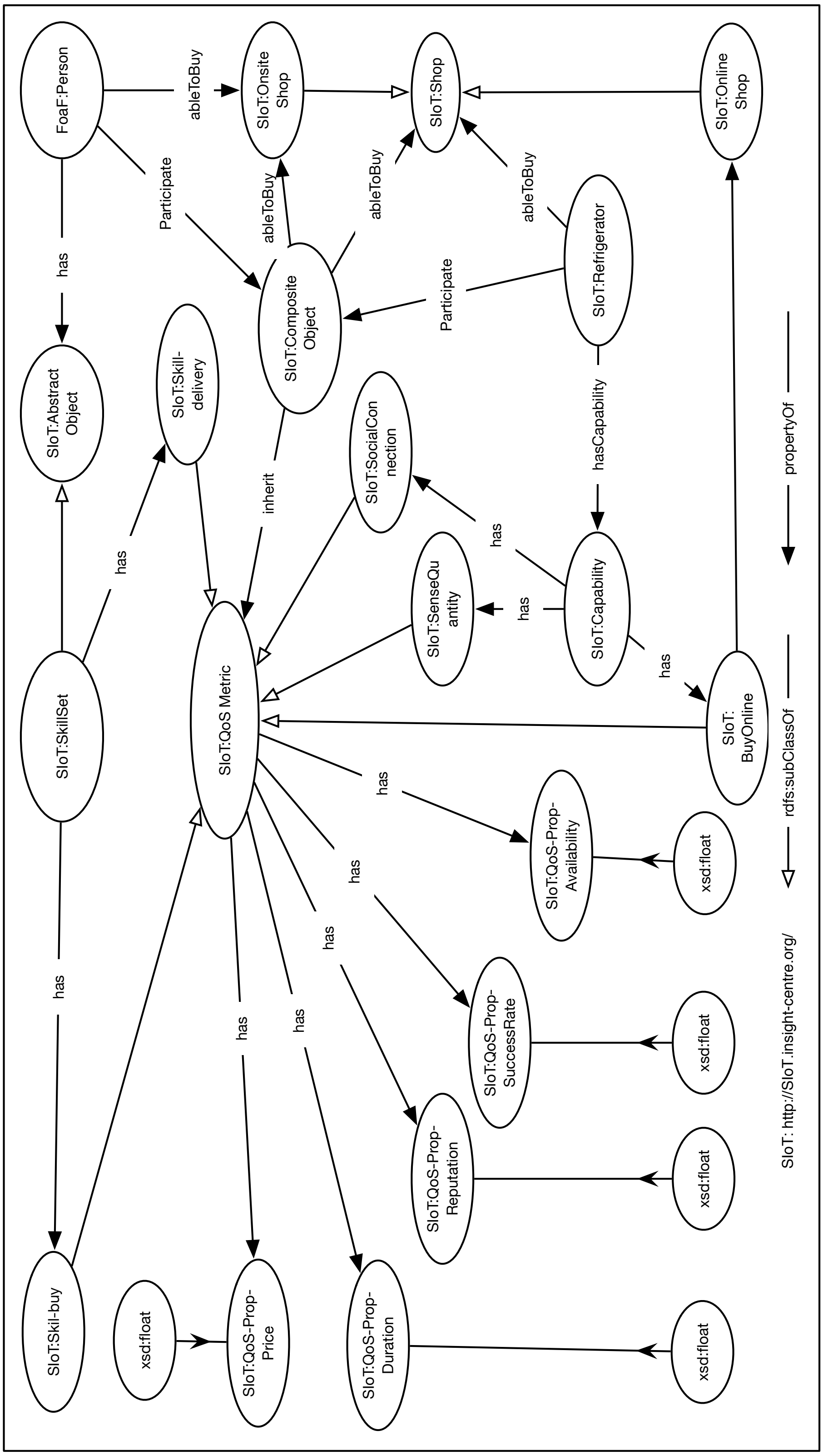

Fig. 4.9 SIoT based smart shopping QoS ontology

- Step-1: Smart Refrigerator detects the necessity to buy milk in 3 days.
- Step-2: Set the Owner's preferences to QoS criteria for buying milk and ask SIoT-MF to considers the following preferences of the owner,
  - Duration = 4
  - Reputation = 3
  - Price = 2
  - Transportation Cost = 1
- Step-3: Sample QoS value from five socially connected smart refrigerators are
  - Duration in days: (2, 3, 4, 1, 3)
  - Reputation score out of 100: (89, 65, 79, 90, 83)
  - Price in Euro: (0.99, 1.0, 0.75, 0.74, 0.75)
  - Transportation Cost in Euro: (1, 0, 2, 0.5, 0.8)
- Step-4: Duration, Price and Transportation cost are negative criteria and Reputation is a positive criteria. SIoT-MF use the technique described in the Chapter-4 and selects the most suitable store and send it to Shopping application.
- Step-5: Shopping application order online if the selected item is available in an online store otherwise it ask SIoT-MF to select a suitable local friend for buying and delivering the item.
- Step-6: SIoT-MF collects Alice's preferences of QoS criteria for this task. The next task is to search for the capable candidates who are willing to buy the item and deliver to Alice's house. The candidate selection method in Chapter 4 is used to select the most suitable candidate from a list of available candidates. Alice's preferences regarding QoS criteria and the QoS value of the available candidates are considered in the selection method. The QoS criteria considered including Alice's preferences are mentioned below,

  Alice's preferences on QoS criteria regarding selecting a candidate object (human):
  - Duration = 5
  - Price = 4
  - Reputation = 3

  - Availability = 2
  - Success Rate = 1

  Sample QoS value from 4 candidate objects

- Step-7: Smart shopping application sets a reminder for Alice's and his friend to remind them about the task. Additionally, the shopping application is able to set GPS location based reminder to both parties as a result it can remind Alice and his friend when they pass by the selected shop. Smart shopping application allows Alice to cancel the agreement only if his friend agrees. Alice and his friend both are capable to change the default reminder settings using the smart shopping application on their devices.
- Step-8: Alice provides two review scores after closing the task , one is to his friend who bought the item for him and another one is to the selected refrigerator who recommended the item. The later one is given after using the item.

There are plenty of purposes can be served by using SIoT-MF, especially where people need a service in a virtual/real-world community. We have described the shopping use case to convey our idea to the readers in a easy way.

4.7 Summary

We have proposed a smart shopping system based on the SIoT-MF to buy necessary items following the recommendation from socially connected friends. In the scenario if the item is available in an on-sie shope then a human is required to buy and deliver the item. In this this case human and physical object participates together to execute a single task. The smart shopping application has two different parts, one is buying ability from online shop and a reminder system to notify human when to buy the item and also to notify when reach near to the selected on-site shop.

In this chapter, we have proposed a framework for Social Internet of Things to enable human-to-physical object communication by creating a virtual counterpart of physical objects in the virtual world. Virtual objects are connected with each other by following semantic ontology model. Composite virtual objects are the next level of the virtual objects which are created according to the application requirements based on the virtual objects and stored in a repository to ensure re-usability and historical analysis. The proposed framework is enforced with real-time processing of structured RDF data and no-RDF data as well. The framework also enables the interaction among objects using social network platforms. This multi-way interaction among human and objects including the human skill that we have identified as an abstract object has a huge potential towards creating a smart society. In the next chapter, we provide the solution for suitable object selection using two different techniques.

Chapter 5

Constraints based QoS aware Object Selection and Composition

In this chapter, we present the object selection process for composing a Composite Virtual Object (CVO). Selecting a suitable object following a user request is not trivial when the number of candidate objects are large in nmbers. Moreover, the QoS of a CVO is a determinant factor to ensure user satisfaction, different user may have different preferences and different requirements. For example, an owner of SIoT-MF based refrigerator scenario (explained in the previous chapter) may wish to minimize the price while satisfying certain constraints for other QoS criteria, while another owner may show more importance on reputation than other QoS criteria. A QoS aware approach to select a candidate object is therefore needed. This approach will increases the QoS of a CVO as well as of the SIoT-MF by taking into account the constraints and preferences of the users.

In the SIoT-MF, the selection process is performed to compose a CVO's based on user requirements as constraints and a set of QoS properties of candidate VOs. The multidimensional QoS model of SIoT-MF includes non-functional properties that are attached to VOs as QoS properties, e.g., availability and reputation. This model destines five QoS properties and ranked by user after completion of a task. IWe cosider two alternative selection approaches to address the object selection issue i.e. Multiple Criteria Decision Making (MCDM) and Integer Programming (IP) solution. MCDM is considered when only one VO is required in a CVO and IP otherwise.

5.1 Contextual Object selection

Decision maker selects a suitable VO from the available pool of candidate VOs. VO-controller creates a new CVO for executing a task. It analyses and scales the QoS criteria of candidate objects. Additionally, the corresponding owner's security, policy and restrictions are also taken into account before selecting an object. Selecting a suitable VO is a complex calculation because of their different types of capabilities and dynamicity.

In this section, we showcase two important phases of VO selection method. Firstly, we represent how we process users request for a task and map this request to identify the most suitable VOs which are capable of performing the desired task. Secondly, we consider that multiple candidate objects are available to perform a task. In such cases, we use Multiple Criteria Decision Making(MCDM) technique or Integer Programming to select a suitable candidate object based on the user's given preferences and below we describe the problem such as; weightage of user preferences,

Notation.

$U = \{u_i, i = 1,2,...,n\}$ is the set of application requirements. $C = \{c_j, j = 1,2,...,m\}$ is the set of a object capabilities. $\xi = \{O_k : k = 1,...,r\}$ is a set of objects. Each object $O \in \xi$ has its set of capabilities $C_O = \{c_j^O \in C : j = 1,...,m\}$. $< u, c >$ iff an user requirements $u \in U$ is matched to a capability $c \in C$.

Definition 1 *Given a set of user requirements U. An object $C \in \xi$ is defined as "a matched capability of an object to U" iff:*

$$|f(U,)| \geq |f(U, C_k)|, \forall C_k \in \xi$$

where $f(U,C) = \{< u, c >: u \in U, c \in O_C\}$, and $|f(U,C)|$ is cardinality of $f(U,C)$

We formalize the object selection process in the Definition 1 and follow the algorithm 1 to find the candidate objects that are capable to execute the task. The selection process is absolutely same for VO and AO where VO has capabilities and AO has skills. However selection criteria are evaluated by their scores given by users in the community. Hence, we have mentioned the object for both VO and AO in the Algorithm 1, and the object capabilities and skills are defined as an object value. The algorithm takes user requirements as input and returns the set of matching objects that are capable to execute the task.

Algorithm 1 Object Selection

Input:User requirements, Candidate Objects=VO, Object capabilities=C
Output: Set of Candidate Objects

```
procedure OBJECTMATCH(M)
    U ← UserRequirements
    O ← Objects
    C ← ObjectCapabilities
    for U ∈ C do
        C ← Object(U, O)
    end for
    return O
end procedure
```

5.2 Object Specification and QoS

SIoT objects create communities according to the social relation of the owner and neighbours. It is important to differentiate the objects in a community with the same capabilities to select the best suitable object. We consider the non-functional properties to select the optimum object, we also adopt an object quality model on a set of quality criteria which are true for all objects. Although we consider a limited number of criteria in this thesis, adding a new criterion will not cause any fundamental change.

In this section, we first present the quality criteria in the context of the VO and CVO. We provide a definition for each criterion first to make it more specific and to show its granularity.

5.2.1 Standalone and Composite Object Specification

A CVO is specified as a composition of VO described following SIoT ontologies which is combined according to the user requirements of the user. It is important to define the integration model and identify the optimal object selection to meet the user constraints and QoS requirements. There are several points that must be considered for CVO composition,

- The set of candidate objectscould be large.
- The quality of a CVO is measured by its end-to-end quality of all VOs rather than any individual VO.
- More than one VOs and CVOs could be available to fulfil a task.

- The selected VOs may continue to evolve with time.
- A task also needs to be capable to accommodate the system variations, such as failure and upgrade.
- An object may have more than one capabilities and their QoS are independent for each capability.
- Service fees need to be mentioned in the object profile for each capability individually.

5.2.2 Quality Criteria for Standalone Objects

We consider five quality criteria for standalone objects. They are price, duration, reputation, success rate and availability. We briefly describe each criteria below.

Price This is the fee that a user has to pay for getting a service from an object. However, the price can be any amount starting from 0 which has to be mentioned in the object profile. We represent the price $q_{pr}(o,c)$, where c is a given capability of an object o

Duration This is the expected time required to execute a task. We compute the duration using

$$q_{du}(o,c) = T_{exec}(o,c) + T_{trans}(o,c)$$

the duration is the summation of the task execution time $T_{exec}(o,c)$ and the transmission time $T_{trans}(o,c)$. Objects shall mention their processing time in their profile or mention a way to ask about it. The transmission time is estimated based on historical performance, i.e.,

$$T_{trans}(o,c) = \frac{\sum_{i=1}^{n} T_i(o,c)}{n},$$

where $T_i(o,c)$ is an incident from the history of transmission time, and n denotes the number of execution times observed in the past.

Reputation Mechanism for measuring the level of trust by user for a specific capability of an object. We define reputation as q_{rep} of an object's capability c. End users experience using the object o is the key factor here and different users experience may vary on the same object as well as on the same capability. The value of the reputation is defined as the average review score given to the capability of an object, i.e.,

$$q_{rep} = \frac{\sum_{i=1}^{n} R_i}{n},$$

where Ri is the end users review score on a capability of an object, n is the number of times the capability was reviewed and scored. Normally, users are asked to choose a score from a given range for reviewing a capability.

Success Rate This criterion is to find that whether a task was completed successfully within the expected time that the object advertised or not. We define success rate q_{rat} of an object's capability c. The value of the success rate is calculated from historical data, by utilizing the expression $q_{rat}(c) = N_c(c)/K$, where $N_c(c)$ indicates the number of successful completion of the task within the advertised expected-time frame, and K is the number of invocations in total.

Availability The availability $q_{av}(c)$ of an object's capability c is that the object is free and ready to do a task. We calculate the value of the availability of an object's capability c by the expression, $q_{av}(c) = T_a(c)$ where T_a is the time the object's capability c is available.

Given the above considerations, the quality vector of a capability c of an object o is defined as

$$q(o,c) = (q_{pr}(o,c), q_{du}(o,c), q_{av}(o,c), q_{rat}(o,c), q_{rep}(o,c)).$$

5.2.3 Quality Criteria for Composite Object

The quality criteria mentioned for stand alone object are also considered to evaluate the composite objects. Table 5.1 shows the aggregation functions.

Price The price $q_{pr}(p)$ of a task having composite objects is based on the composition plan p, p_r is a composite value of the prices of the object's capabilities that participate in p. Table 5.1 shows the equation for the execution price, $c(t_i)$ denotes the capability invoked by task t_i.

Criteria	**Aggregation function**
Price	$q_{pr}(p) = \sum_{i=1}^{N} q_{pr}(o_i, c(t_i))$
Duration	$q_{du}(p) = CPA(p, q_{du})$
Reputation	$q_{rep}(p) = \frac{\sum_i^N q_{rep}(o_i)}{N}$
Success rate	$q_{rat}(p) = \frac{\sum_i^N q_{rat}(o_i)}{N}$
Availability	$q_{av}(p) = \sum_{i=1}^{N} q_{av}(o_i, c(t_i))$

Table 5.1 Aggregation function for calculating the QoS values for Composite Objects

Execution duration The duration $q_{du}(p)$ of a composition plan p is the sum of all the VO's duration in a CVO.

Reputation The reputation $q_{rep}(p)$ of a composition plan p is the average of the reputations of the capability of the object that participate in the plan p.

Success rate The success rate $q_{rat}(p)$ of a composition plan p is the average of the success rate of the capability of the object that participates in the plan p.

Availability The availability $q_{av}(p)$ of a composition plan p is defined by the availability of all the required capabilities of the objects that participate in the plan p

Given these functions, the quality vector of a composite object's is defined as

$$q(p) = q_{pr}(p), q_{du}(p), q_{rep}(p), q_{rat}(p), q_{av}(p)$$

5.2.4 Multiple Constraint Decision Making (MCDM)

We use the Multiple Criteria Decision Making (MCDM) [69, 206] technique to select a suitable candidate VO. In this process, we consider the weight of QoS criteria given by a user as preferences. We consider five QoS criteria, however, any new criteria may also be included without any fundamental changes in the calculation. For example, we consider five criteria of SIoT objects along with a user-defined weight, numbered from 5 to 1 where higher is the most desired by the user. 5 = Price 4 = Reputation 3 = Availability 2 = Success rate 1 = Experience. Given a requirement in the VO controller, there is a set of candidate VO and AO, $Oj = O1j, O2j, ...Onj$, that can be selected to execute the task. By merging the quality vectors of all the candidate VOs and AOs a matrix

$F = (Fi, j; 1 <= i <= n, 1 <= j <= 5)$ is built, where each row Fj corresponds to a VO and each column corresponds to a quality dimension. Some criteria are positive i.e., the higher the value, the higher the quality i.e. Reputation, Availability, Success rate and Experience. On the other hand, there are negative criteria also i.e., the higher the value, the lower the quality e.g. Price. Positive criterion values are calculated with equation 1 and negative criteria values are aligned to equation 2,

$$V_{i,j} = \begin{cases} \frac{F_{i,j} - F_j{}^{min}}{F_j{}^{max} - F_j{}^{min}} & \text{if } F_j^{max} - F_j{}^{min} \neq 0 \\ 1 & \text{if } F_j^{max} - F_j{}^{min} = 0 \\ \end{cases} \tag{1}$$

$$V_{i,j} = \begin{cases} \frac{F_j{}^{max} - F_{i,j}}{F_j{}^{max} - F_j{}^{min}} & \text{if } F_j^{max} - F_j{}^{min} \neq 0 \\ 1 & \text{if } F_j^{max} - F_j{}^{min} = 0 \\ \end{cases} \tag{2}$$

Where F_j^{max} is the maximum value of a quality criteria in the matrix F i.e. $F_j^{max} = Max(F_{i,j}), 1 \leq i \leq n$ while F_j^{max} is the minimal value of a quality criteria in the matrix F i.e. $F_j^{min} = Min(F_{i,j}), 1 \leq i \leq n$ while F_j^{max}, Hence we set a matrix $V = (V_{i,j}; 1 \leq i \leq n, 1 \leq j \leq 5)$ in which each row V_j corresponds to a object $O_{i,j}$ while each column corresponds to a quality dimension.

Weighting phase

Now we calculate the score of each VO using the following formula, $Score(s_i) = \sum_{j=1}^{5}(V_{i,j} * W_j)$, where $W_j \ \epsilon \ [0,1]$ and $\sum_{j=1}^{5} W_j = 1$. W_j represents the weight of criterion j. Users express their preferences according to the weight of each features. Finally based on the score the best VO is selected. If no VO matches with the requirements for a given task, a notification is generated to relax these constraints or waits for a capable VO.

5.2.5 Integer Programming

The MCDM approach may seem inefficient sometime when there are n tasks and m candidate objects for each task and the total number of composition plan is m x n. Moreover, MCDM do not consider object constraint which may select an object who has dependencies or involvement with other objects. To overcome this problem we propose a method based on Integer Linear Programming (IP) solution for selecting a suitable object.

5.2.6 Integer Programming (IP) Solution

In this section, we use the IP solution approach for selecting the most suitable object. We map the problem of selecting a suitable composite object with the IP problem in following steps;

For every object's capability c_{ij} that can be required by a task t_j, we involve an integer variable y_{ij} in the IP problem, conventionally y_{ij} is 1 if capability c_{ij} is selected for task t_j, 0 otherwise [206]. A set of integer variables x_j is included, hence, x_j defines the expected start time of task t_j. The constraints of execution duration will use this set of variables. Now, we use the following function to calculate the score,

$$Max\left(\sum_{l=1}^{2}\left(\frac{F_l{}^{max}-F_{i,l}}{F_l{}^{max}-F_l{}^{min}}*W_l\right)+\sum_{l=3}^{5}\left(\frac{F_{i,l}-F_l{}^{min}}{F_l{}^{max}-F_l{}^{min}}*W_l\right)\right) \tag{3}$$

Where F_l^{max} is the maximum value of a quality criteria in the matrix F i.e. $F_l^{max} = Max(F_{i,l}), 1 \le i \le n$ while F_l^{min} is the minimal value of a quality criteria in the matrix F i.e. $F_l^{min} = Min(F_{i,l})$. Now, $W_l \, \epsilon \, [0,1]$ and $\sum_{j=1}^{5} W_j = 1$, W_l is the weight assigned to the QoS criteria.

There is a set of candidate objects O_j available for each task t_j. Nevertheless, we need to select one object to execute the task t_j. Provided that, y_{ij} denotes the selection of object o_{ij} to execute task t_j, the following constraints must be satisfied:

$$\sum_{i \epsilon O_j} y_{i,j} = 1, \forall j \epsilon A, \tag{4}$$

A is the set of tasks, e.g., 1000 potential candidate objects are available that can execute task j. but, only one of them need to be selected for executing task j, we have that

$$\sum_{i=1}^{1000} y_{i,j} = 1.$$

Constraint on Execution Duration Let x_j denote the expected start time of task t_j, p_{ij} denote the duration of task t_j when assigned to object o_{ij}, and p_j denote the expected duration of task t_j knowing which object has been assigned to it. We have the following constraints:

$$\sum_{i \epsilon O_j} p_{ij}\ y_{ij} = p_j, \forall j \epsilon A, \tag{5}$$

We can see in constraint (3) that the duration of a given task t_j has to be the duration of one of the objects in A, because one and only one of these objects can be selected to execute task t_j.

Also, let $t_j \to t_k$ denote the fact that task t_k is a direct successor of task t_j. Now, constraint (4) shows that if task t_k has a direct dependency on task t_j, then the execution of t_k must be initiated after the completion of task t_j.

$$x_k - (p_j + x_j) \geq 0, \forall t_j \to t_k, j, k \epsilon A, \tag{6}$$

We can see in the constraint (5) that a composite object can only be treated as completed only when all the participated objects in the composite objects complete their task.

$$q_{du} - (x_j + p_j) \geq 0, \forall j \epsilon A. \tag{5}$$

Success rate and reputation are calculated by users given a score (see Table 5.1) and the parameter of the criteria for price and availability set by the object owner, as a result, no constraint is required.

It is worth mentioning here that mapping object's service selection with IP problem is generic and we have considered five criteria, however adding new criteria will not include modification in the core system.

5.3 Experimentation

To measure the performance of the object selection techniques used in the SIoT-MF we have conducted a pragmatic evaluation by varying the number of users, available candidate objects .i.e., VO and number of VOs in a CVO i.e., size of a CVO.

The dataset contains 10 million objects, 100 thousand CVOs and 500 users with a random weight of user preferences on 5 QoS criteria in each request. The synthetic dataset are generated in Java language, the value were inserted to each VO randomly along with

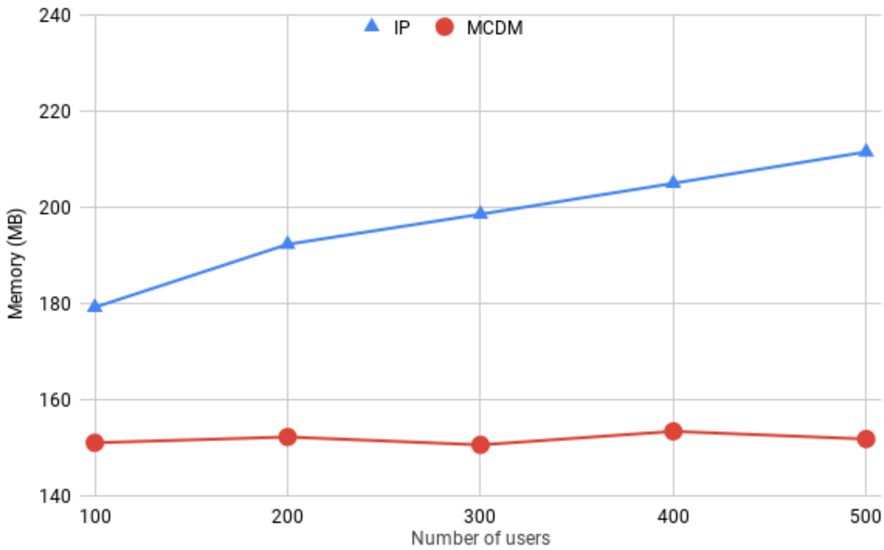

Fig. 5.1 Cost of memory with increasing number of users while number of candidate VOs and size of each CVO were static

sequential unique identification number. Additionally, unique identification number was assigned to each user as well. We generated a jar file from the source code and conducted the experiment by changinf different paramter from the command prompt. The entire experiment was conducted using both Integer programming (IP) and Multiple Criteria Decision Making (MCDM) techniques to observe their behaviour in different settings.

We have measured the memory consumption and latency as the performance metric. The amount of memory was consumed and the amount of time was required to select suitable VOs to create the CVOs are defined as memory consumption and latency respectively. Each experiment was conducted 10 times and we have taken the average value of each experiment. The experiment testbed is a PC environment that includes Mac book Pro, processor 2.6 GHz Intel Core i5, Memory 8 GB 1600 MHz DDR3.

5.3.1 Effects of increasing number of users

The intention here is to see the effects on the systems and the behaviour of two techniques MCDM and IP when the number of users increases gradually. We compare both techniques regarding the memory consumption and latency by varying number of users, started from 100, 200, 300, 400 and 500 while other parameters were stable i.e candidate VOs were 30,000 and each CVO contains 3 VOs.

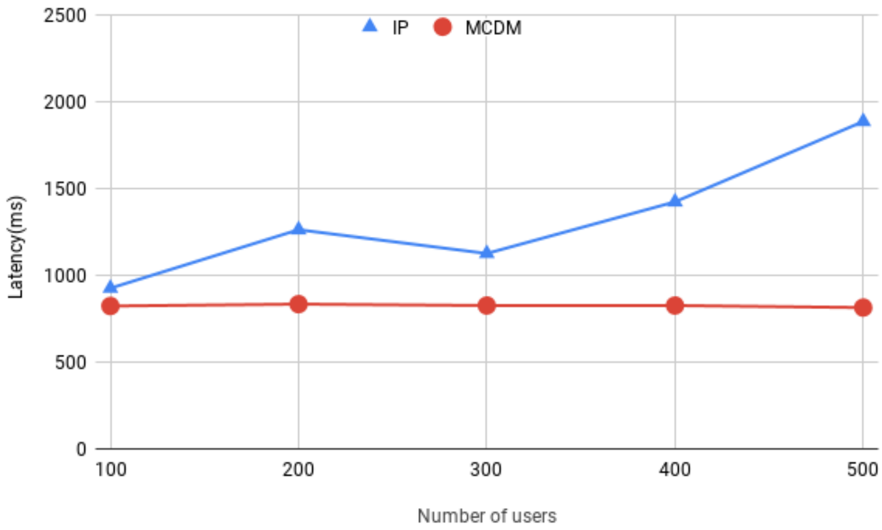

Fig. 5.2 Latency with increasing number of users while number of candidate VOs and size of each CVO were static

Figure 5.1 shows a comparison of the memory consumption of MCDM and IP in the systems. MCDM do not consider constraints at the object level in the selection process, as a result, it shows a stable line in terms of memory consumption. However, IP considers several constraints for selecting a candidate object from the poll of candidate objects. In the Figure 5.1 we see that when the number of users increases it consumes more memory than MCDM and the consumption goes higher when the number of users increases. Figure 5.2 depicts latency of IP and MCDM, we use the same settings as the memory consumption experiment and observe a similar kind of behaviour of the two techniques. However, the performance difference between MCDM and IP is tolerable.

5.3.2 Effects of the size of CVOs

CVOs are formed according to user requirements and that is why the number of VOs in a CVO is a variable. The reason for this experiment is to observe the behaviour of the systems when the number of VOs are increased in a CVO. Latency and memory were measured and compared w.r.t. MCDM and IP by gradually increasing the number of VO in a CVO while the users are 10 and candidate objects are 1000. Figure 5.3 presents that IP consumes a slightly more memory than MCDM when CVO size increases. The latency of the systems in this settings is presented in the Figure 5.4, we can see IPs latency got a pick when 40 VOs were required in a CVO, the reason is IP consider constraints and sometimes

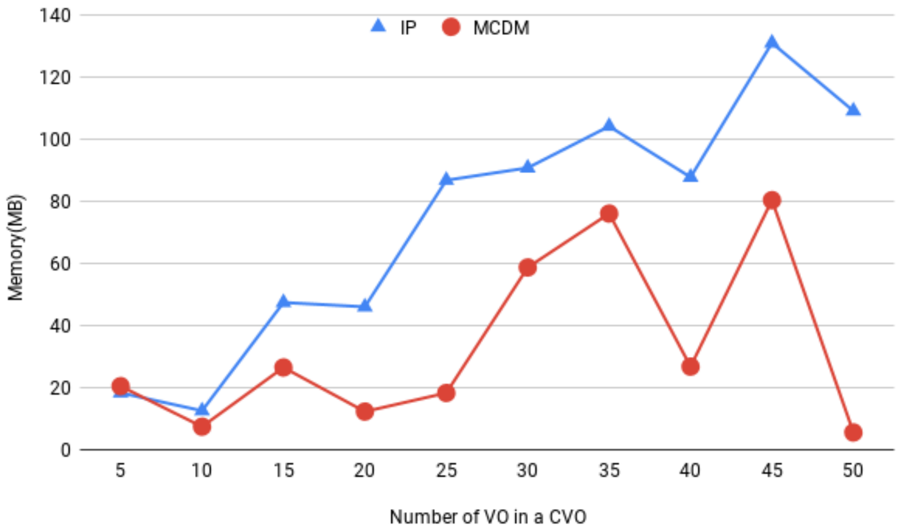

Fig. 5.3 Memory Consumption when CVO size increases gradually

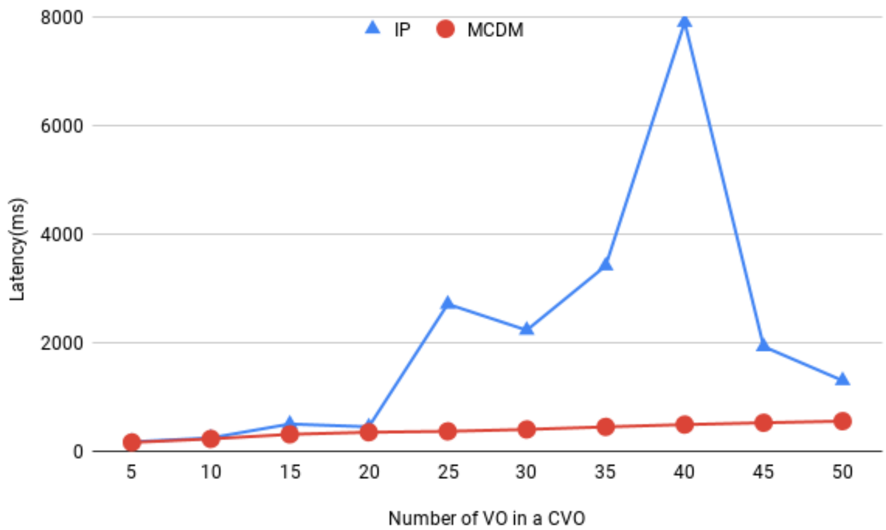

Fig. 5.4 Latency when CVO size increases gradually

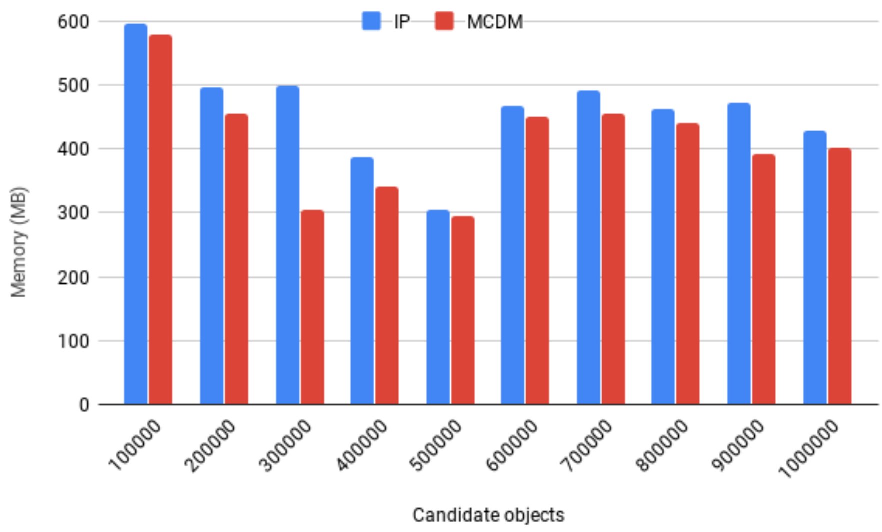

Fig. 5.5 Memory Consumption with increasing number of candidate objects

it has to search again and again for the desired object which requires more time, however MCDM do not consider constraints and may select an objects which have dependency on other other objects.

5.3.3 Effects of the increasing number of candidate objects

Size of elements is trivial to measure the efficiency of a selection technique. In this experiment, we measure the performance in terms of memory consumption and latency of the systems using MCDM and IP. We have varied the number of candidate objects from 100000 to 100000 gradually and kept other parameters at a stable state i.e number of users were 10 and number of VOs in a CVO were 3. Figure 5.5 and Figure 5.6 show that IP consumes slightly more memory and time than MCDM when the number of candidate objects changes which we believe is ignorable to select IP over MCDM when more than on VO is required in a CVO.

It was expected that IP will cause more computational cost than MCDM as IP consider constraints at the candidate object level. However, we intended to check the difference in computational cost when using IP and MCDM. According to our experiment in most of the cases, IP consumes slightly higher memory and time which is ignorable

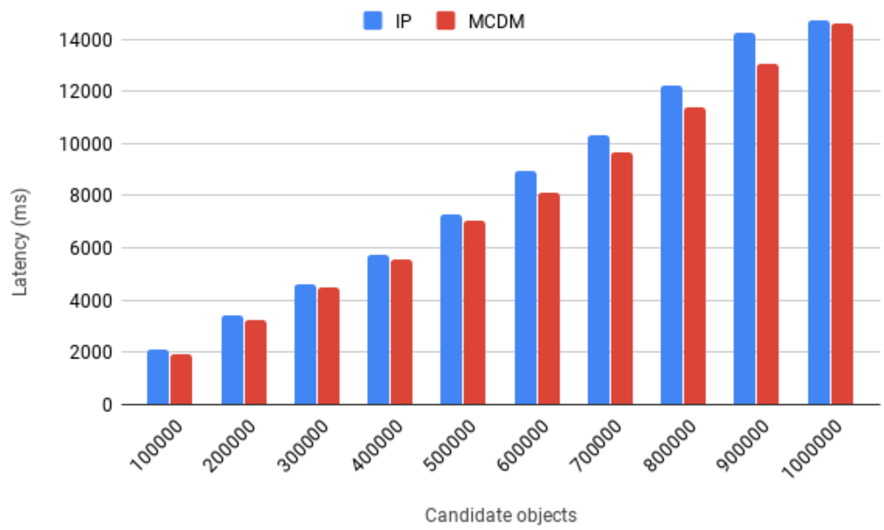

Fig. 5.6 Latency with increasing number of candidate objects

5.4 Summary

In this chapter, we explained constraint-based QoS aware object selection technique in the SIoT-MF. We provide two solutions to address a suitable object selection problem are, a)MCDM and b) IP. The MCDM may not be the best solution when the candidate objects are huge in numbers or when participating VO in a CVO has a dependency on each other. As a result, IP solution is alternate option to select a suitable object. We have considered the QoS criteria for each object's capabilities during the selection procedure. Our experiment shows that choosing IP over MCDM will not cause any big difference in computational cost. As a result, we keep both techniques and use MCDM when only one VO is required or a number of candidate objects are limited and has no dependencies with each other.

Chapter 6

Adaptive Real-time Analysis and Monitoring SIoT Objects

In the previous chapters (Chapter 3 to Chapter 4) we have depicted the SIoT-MF concept, design, real-world object virtualization and object selection method in the SIoT-MF. The virtual representation of the physical objects i.e. the VO communicates with the physical counterpart i.e. the real-world SIoT objects generate data continuously. To cop-up with these continuous data we refer to real-time processing mechanism in the SIoT-MF that ensure data freshness as well as ensure the real-time processing capability in the framework. The term data freshness implies that the most recent data are more meaningful.

Since SIoT-MF offers semantic ontology and the QoS ontology of VO and semantic continuous data of VO are processed and queried by RDF Stream Processing (RSP) engines, the performance of the engines has a great impact on the performance of SIoT framework as well as on the user experience.

Several RDF Stream Processing (RSP) engines are available to query semantic data streams. We have the freedom to select a most suitable RSP engine following the VO controller's requirements. This selection needs to be done at design time resulting in early bound rigid solutions that are unable to adapt to changing requirements.

We propose an adaptive approach to comply with the changing requirements from the VO controller. VO controller follows the user requirements to create VO and CVO. As a result, it is important to adapt to dynamic user requirements during run-time. We introduce an adaptive layer which supports late binding approach regarding the selection of RSP engines and results in an efficient processing of a larger amount of stream data from real-world SIoT objects. Moreover, the adaptive layer includes input and output control, monitoring the status of the underlying RSP engines, as a result, the adaptive layer is essential even if there is only one type of RSP engine's multiple instances are available. As a proof of the necessity of an adaptive approach, we evaluate two most popular RSP engines and point out the effects of different run-time features.

We understand that different RSP engines are proposed with a different understanding of RDF stream and we also believe that combining their strengths under a single framework is essential from the users' point of view. In this way, their differences can be used as combined strengths rather than competing with each other. We propose an adaptive layer on top of existing RSP engines after analysing their strengths and weakness at a granular level to come up with an efficient solution under a single flexible framework. We have mainly addressed the following issues in this chapter,

- We identify key requirements to highlight the need for adaptivity.
- We categorise different parameters and features of RSP engines and showcase how dynamic changes in requirements can be handled by adapting to different settings of RSP parameters.
- We evaluated the performance of existing two RSP engines to prove the need for adaptivity.

In summary, this chapter describes the real-time processing mechanism of SIoT-MF. Additionally, it investigates the strengths and weakness of the two popular engines i.e., CQELS Le-Phuoc et al. [117] and CSPARQL [31] and finally offers an adaptive approach for selecting one engine following the application requirements [167, 169].

6.1 Features of Stream Processing Engines

Over the last few years, several RSP engines have been proposed for efficient processing of RDF streams [31, 117, 111, 14, 42, 37, 60]. However, few of them are still in their cradle and are not published with the deployable resources. Hence, the most popular and deployable RSP engines i.e. CSPARQL, CQELS are affected and differ by multiple aspects, including the execution method, input data model, query language, operational semantics, output streaming operators, execution time, processing techniques. Considerable manual efforts go into creating and tuning such diverse and dynamic engines. In addition to that, VO controller and user requirements, input data from PO and workload properties of RSP

Engine \ Feature		Execution	Output	ERN	Query Language
CQELS		Data	I	No	CQELS native
CSPARQL		Time	R	Yes	CSPARQL native

Table 6.1 Comparison of CQELS and CSPARQL w.r.t. RSP features

Legend: Execution=Query Execution strategy, Output=Output streaming operators, ERN=Empty Relation notification, Time=Time driven, Data=Data driven, I=Istream, R=Rstream, D=Dstream, No=Not supported, Yes=Supported

engines may change over time, often in unpredictable ways. As a result, a mechanism to adapt to this diversity is required in order to satisfy user with a best possible way of changing environments by considering the user requirements and available resources of RSP engines. Due to this diversity, it is hard to select a single RSP engine for any particular application. Moreover, few of the RSP engines capabilities are limited due to the decision made during the design time of any RSP engine, which can not be changed during the lifespan of a query engine. Existing approaches to stream query processing mostly offer rigid solutions and lack the adaptability to accommodate the changing requirements of the applications and properties of the underlying data streams.

Our analysis shows that RSP engines differ in various features, we classify these RSP features into two main categories:

i) *design-time features* include aspects such as input data model, query language, execution strategy, supported output streaming operators, and

ii) *run-time features* include aspects such as execution time, time window size, input stream rate and available resources. The calculation of these features is explained in the section 6.4.2.

CQELS and CSPARQL are taken into consideration to analyse their available features and then narrowed down to Table 6.1.

Execution Strategy Query execution in RSP follows two types of strategies i.e. Time-driven, we denote as X_t and Data-driven, we denote as X_d. In the Time driven strategy, query execution follows a periodic schedule, and the Data-driven strategy triggers query execution when new stream data arrives at an RSP engine. Both strategies have their pros and cons such as Time-driven execution may not be suitable where stream arrival rate is higher than the query execution and the output result may compromise data freshness. The data-driven strategy may not be suitable where stream arrival rate is very high, but the output result is not required with that rate. Moreover, in Data-driven strategy, high stream arrival rate refers to a higher number of query execution that consumes more resources. As a result, the RSP engine may get overloaded because of some unnecessary resource consumption. We denote data-driven as X_d.

Output Operators Output operators are required to form a new stream from the output of the query. They are also known as the Relation-to-Stream (R2S) operators[15, 31, 42]. R2S contains three different operators ***Rstream*** *(O_r) includes the entire output relation in the stream,* ***Istream*** *(O_i) includes only the new arrivals added to the relation* and ***Dstream*** *(O_d) includes only those which are deleted from the relation*

Empty Relation Notification (ERN) This is another R2S operator. It notifies if there is no mapping relation after executing a query. This feature lets VO controller know that there are no desired results but streams are coming from the PO. This feature is important for VO controller as it can decide whether to wait or select another PO. When this facility is available in an RSP engine we denote it $ERNy$ and when it is not available in an RSP engine we denote it $ERNn$.

Query language: Each engine has their own native query language which is a great challenge to decide which query language should be followed for which requirements. To overcome this issue the adaptive layer is capable of transforming the query according to the selected engines query language.

Features	**Type**	**Description**	**Denotation**
Execution	Time	follows a periodic schedule	X_d
	Data	when new stream data arrives	X_t
Output	I	includes only the new arrivals added to the relation	O_i
	R	includes the entire output relation in the stream	O_r
	D	includes only those whcih are deleted from the relation	O_d
ERN	Yes	notifies if there is no mapping relation	ERN_y
	No	do not provide notification	ERN_n

Table 6.2 Narrowed down design time RSP features and denotation

We consider the aforementioned RSP features that have an impact on engines performance. We summarise all the features in the Table 6.2. It is worth mentioning here that in this chapter we have used CQELS and CSPARQL because of their better performance have been presented in several Benchmarks [208, 61, 116].

In the Adaptive approach, a matchmaking is required to find the best RSP engine for a given set of requirements and a set of RSP features. We formalise this Matchmaking process.

Definition 2 (Matchmaking) *This definition formalises the Matchmaking among application requirements and RSP capabilities.*

Let $f()$ *be the adaptive RSP function for finding the best solution based on application requirements set A and RSP capability set C,*

$A = \{a_i, i = 1,2,...,n\}$, *where* a_i *be an application requirement and A be the set of application requirements*

$E = \{C_i, i = 1,2,...,n\}$, *where* C_i *be an RSP capability and E be the set of RSP engine's capabilities,*

$\xi = \{E_i, i = 1,2,...,n\}$, *where* ξ *be the set of RSP engines and* E_i *be an RSP engine where each* E_i *contains C*

Hence $f(A,\xi) \rightarrow E$ *such that,* $E \cap A \neq \phi$ *; function* $f(A,\xi)$ *finds the match between application requirement and RSP engine's capabilities and select the best engine, where no match is undesirable.*

6.2 Validating the Necessity of Adaptivity

To showcase the need of an adaptive layer for RSP and focus the impact of different design-time and run-time RSP features we have conducted an empirical evaluation over two RSP engines i.e. CQELS and CSPARQL. We set-up our testbed using CityBench datasets [9], which are based on real datasets collected from the City of Aarhus, Denmark. CityBench also provides a set of queries as part of the benchmark, we have used three CityBench queries. It is worth mentioning here that, we do not have real-world SIoT data in our hand so that we decided to use CityBench data to test the RSP engines. However, it does not make any difference at the performance level. We intend to evaluate RSP engines behaviour in terms of latency, memory and also to find the breaking points where RSP engines stop producing results. We evaluated the run-time features of RSP engines in different settings to see the effects of runtime features on the performance of RSP engines. All the experiment conducted on PC environment in a Mac book Pro, processor 2.6 GHz Intel Core i5, Memory 8 GB 1600 MHz DDR3. We evaluate the following features;

6.3 CityBench Dataset

Thanks to the CityPulse [1] project and CItyBench [9] benchmark, they have made the data available for public use. All the data are collected from the city of Aarhus, Denmark. We provide a brief description of the dataset below based on CityBench and CityPulse,

Vehicle Traffc Dataset. These are traffic data from 449 sensors deplyed by city administrator. time and congestion level between the two points set over a segment of road.

Parking Dataset. Parking management system in Aarhus provides live data streams and indicates the number of vacant parking spaces. Observations are generated from 8 parking lots. This data is available from a RESTful web service5 in XML format.

Weather Dataset. Weather sensor data in the city provides observations related to dew point(C), humidity(%), air-pressure (mBar), temperature (C), wind direction and wind speed (kph).

Pollution Dataset. Pollution has a direct relation to the traffc level but due to the unavailability of pollution sensors, a synthetic dataset were provided. Details of the procedure followed to synthesised pollution data can be found at: [2]

Cultural Event Dataset. This dataset contains cultural events provided by the municipality of Aarhus. The dataset is periodically updated to reflect the latest information related to planned cultural events. Updates can be delivered in data streams (a notifcation

[1]http://www.ict-citypulse.eu/page/
[2] http://iot.ee.surrey.ac.uk:8080/datasets/pollution/readme.txt.

service notifies of any updates in the dataset) or it can be used as background knowledge that updates on a daily or weekly basis.

Library Events Data. This dataset contains a collection of past and future library events hosted by libraries in the city. A total collection of 1548 events is described in this dataset. Similarly to the Cultural Events Dataset, updates in the

Library Events Dataset. are also not frequent, therefore, the dataset is considered quasi-static. This dataset can be crawled from the city library website8.

User Location Stream. Most of the IoT-enabled Smart City applications are designed to be location-aware, therefore they strongly rely on updates on the location of mobile users. A User Location Stream is simulated to mock-up the movements of a user.

Effects of input stream rate on the performance of RSP engines

We intend to see the effect of different input stream rate in CQELS and CSPARQL in terms of latency and memory. We controlled the delay of input stream at a different frequency in millisecond i.e 200, 600, 800, 1000, 1200, 1400, 1600, 1800, 2000. We expect that memory consumption and latency will be different at each input rate when all other features are identical. Theoretically, CQELS does more execution than CSPARQL as CQELS executes the query at each data arrival, on the other hand, CSPARQL executes at the end of each time window. Hence, there should be a difference in memory consumption and in latency according to the number of executions and the buffering till the end of the time window. This experiment will help to show that switching among engines according to the run-time and design-time features are not trivial and also smartly controlling the input rate and monitoring the status of RSP engines help to protect them against breaking in a production environment.

Effects of Concurrent queries on the performance of RSP engines

Processing of concurrent queries in CQELS or CSPARQL demand more memory and include processing delay. Hence we have registered concurrent queries to both engines by duplicating a query 20 times, 50 times and 100 times. In reality, RSP engines may need to process thousands of concurrent queries, the intention is to see which engine performs better with concurrent queries and at which point it breaks or stop responding. This experiment will also help to prove the necessity of an adaptive layer where an adaptive layer will keep the track of registered queries and the status of RSP engine's free memory.

Effects of Time Window size in RSP engines:

According to the design architecture CQELS and CSPARQL have two different time window mechanisms. CQELS does not have to buffer the data for a long time when the input rate is very fast i.e input stream delay is lower as it executes queries immediately when a stream arrives. On the other hand, CSPARQL holds the data till the time window finishes. As a result, CSPARQL executes a query even if there is no input stream or it will wait till the time window ends regardless of the arrival of data. These two different mechanisms have pros and cons; our intention is not to criticize them rather our evaluation will figure out their effect on memory and latency in different settings. We need to find the trade-off between CQELS and CSPARQL to select an engine at a given situation. In this experiment, we will use two different sizes of time window with concurrent queries and different input rates as well.

6.3.1 Latency

The amount of time required from the arrival of an input stream to producing output result stream at an RSP engine is defined as latency. We have investigated the latency by changing several run-time features i.e. varying the input stream rate, registering concurrent queries and changing the time window size. We have considered most of the combination of these three features with different parameter values.

Figure 6.1 to Figure. 6.10 depict the latency of Query1 for CQELS and CSPARQL. We observed that CSPARQL never responded when the input stream delay is 200ms to 600ms but CQELS seems consistent at lower input stream delay. Interestingly both latency lines in Figure 6.2 to Figure 6.10 crossed each other at several points. CSPARQL have lower latency with 20 and 50 concurrent queries in Figure 6.2 and Figure 6.3 but with 100 concurrent queries in Figure 6.4 none of them is stable and have crossed each other at every alternate steps.

Changing the window size also have an effect on latency as depicted from Figure 6.5 to 6.10. We have used three sets of windows in this experiment i.e. default CityBench query's

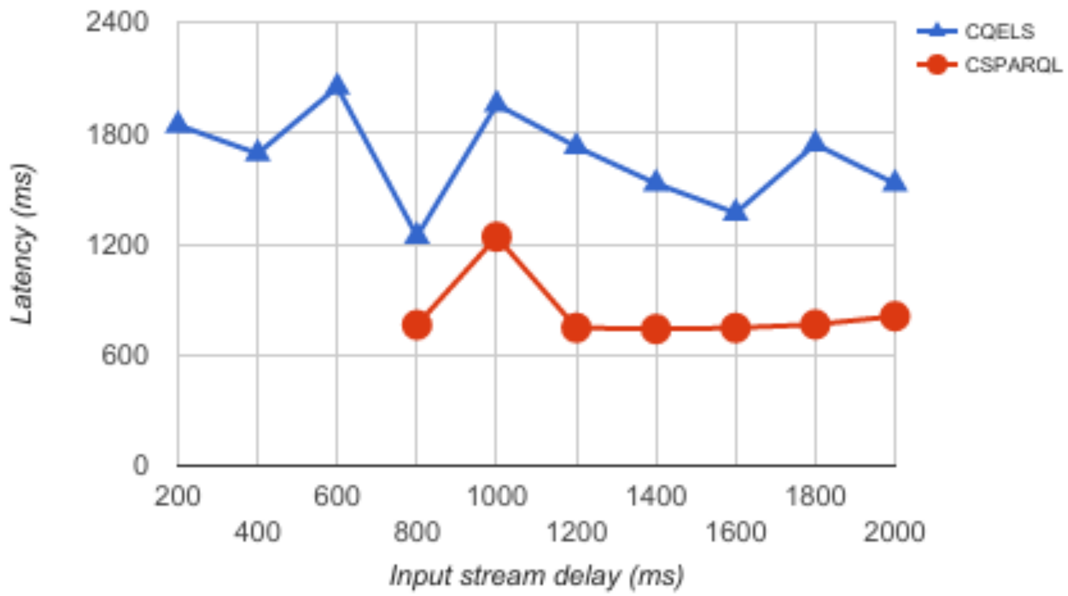

Fig. 6.1 Q1 over various input stream rate

windows, 3 times larger than the default windows (larger time window) and then three times smaller (smaller time window) than the default windows.

The larger window causes decrement in latency for both engines in Figure 6.5, 6.6 and 6.7 compare to the CityBench default windows in Figure 6.1, 6.2 and 6.3. Interestingly CQELS follows a down trend with large windows compare to CSPARQL after consuming more time than CSPARQL till 1800 ms. However, with the 50 concurrent queries in Figure 6.7 it takes a rise again. Figure 6.8, 6.9 and 6.10 show that smaller window increase the latency in both engines however with 100 concurrent queries the latency is almost same to default window query.

It is worth mentioning here that we have used other two queries but both the engines stop responding after a period for CityBench Query 4, and for Query6 only CQELS responded as a result latency cannot be compared for these two queries. However, we have evaluated the memory consumption of all three queries in the next section.

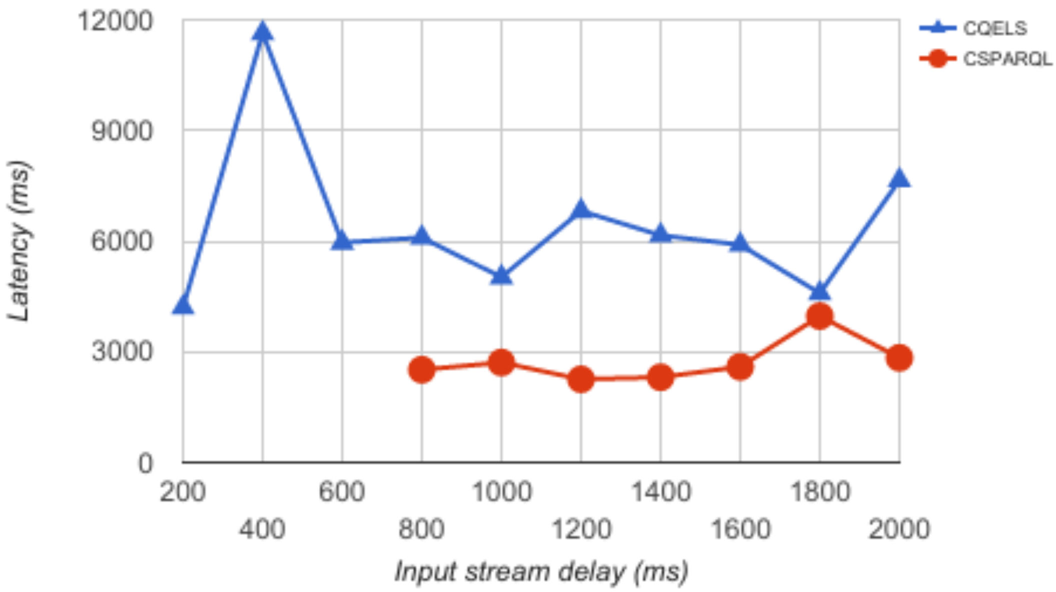

Fig. 6.2 Q1 over various input rate with 20 concurrent queries

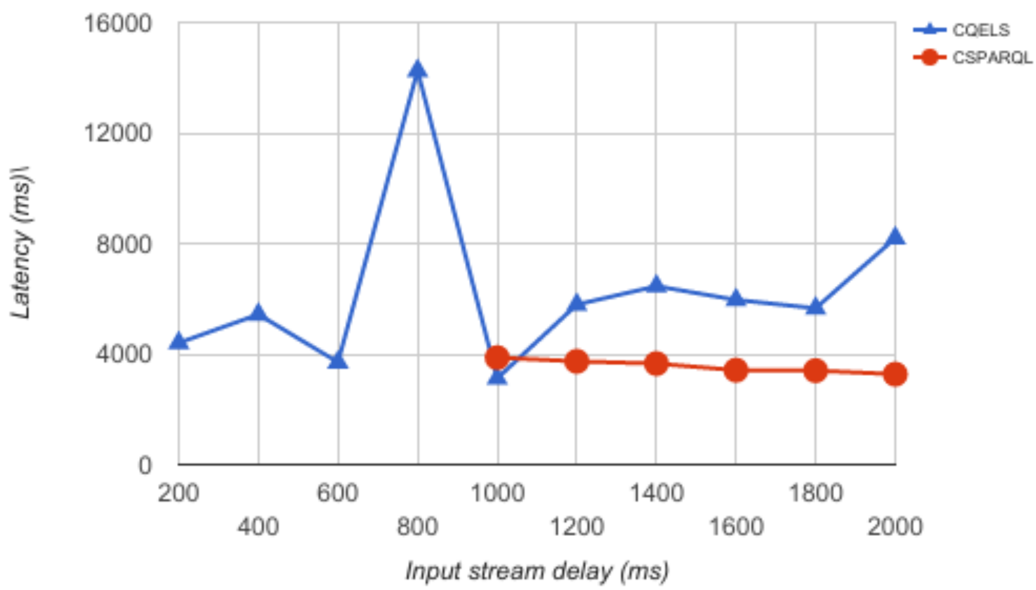

Fig. 6.3 Q1 over various input stream rate with 50 concurrent queries

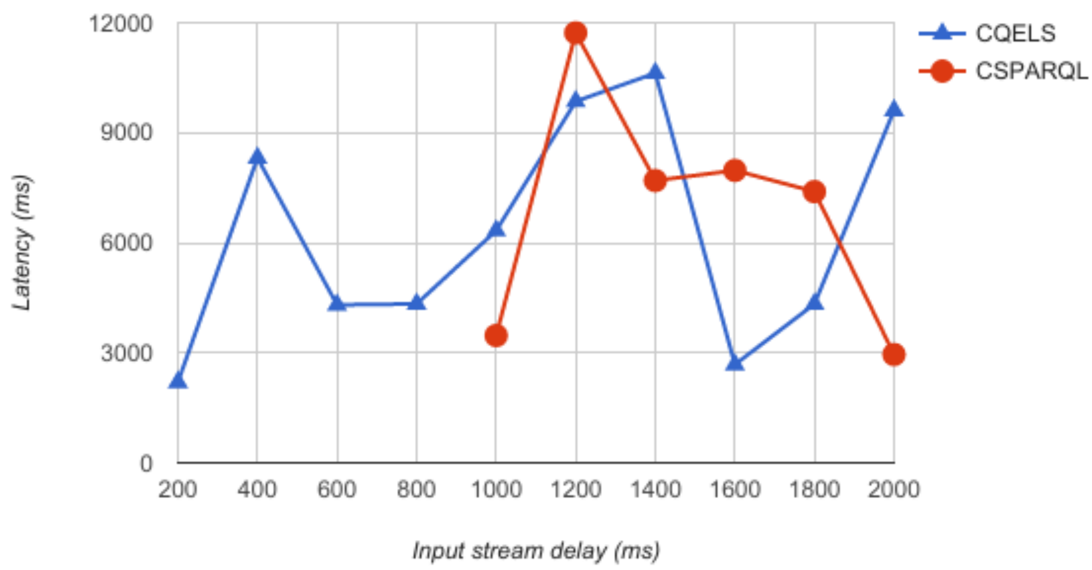

Fig. 6.4 Q1 over various input rate with 100 concurrent queries

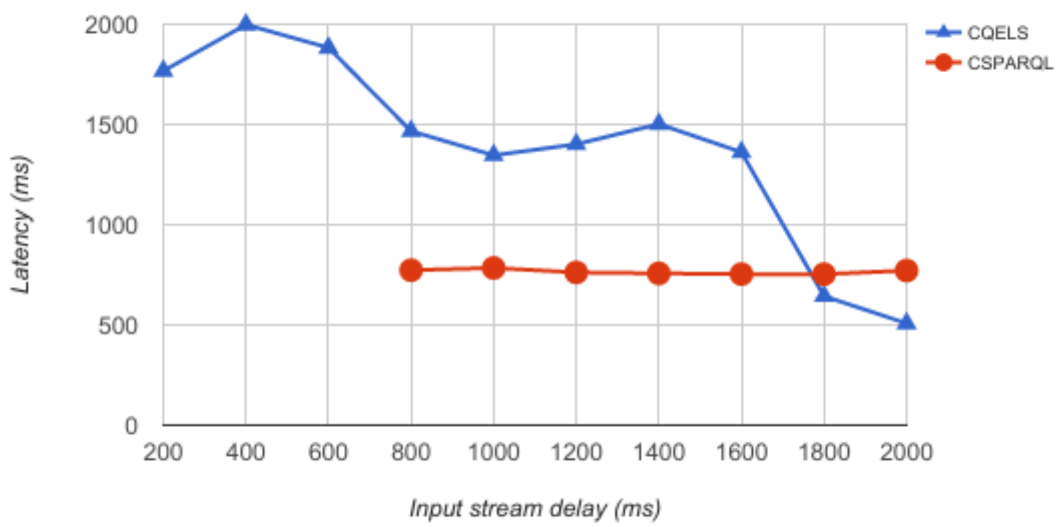

Fig. 6.5 Q1 over various input stream rate with large time windows

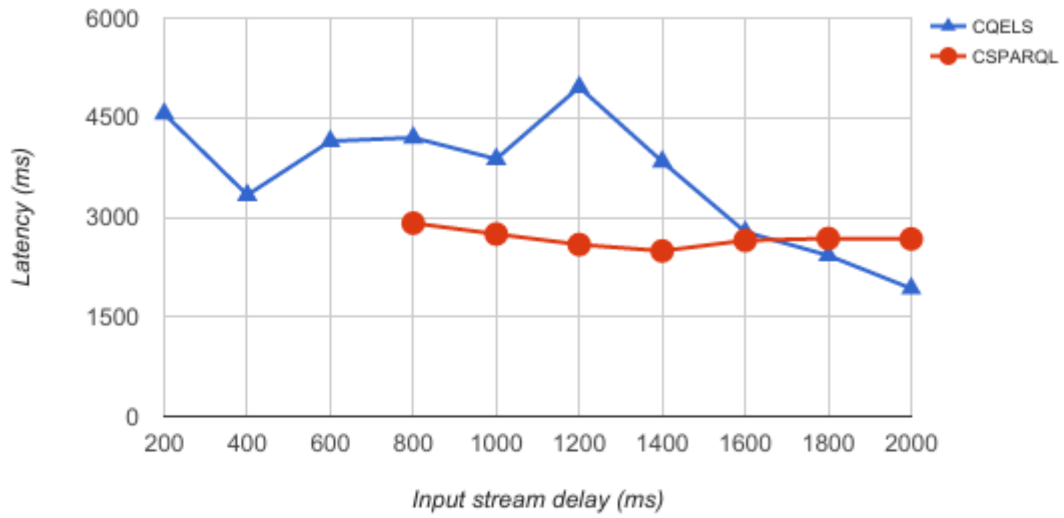

Fig. 6.6 Q1 over various input rate with large window and 20 concurrent queries

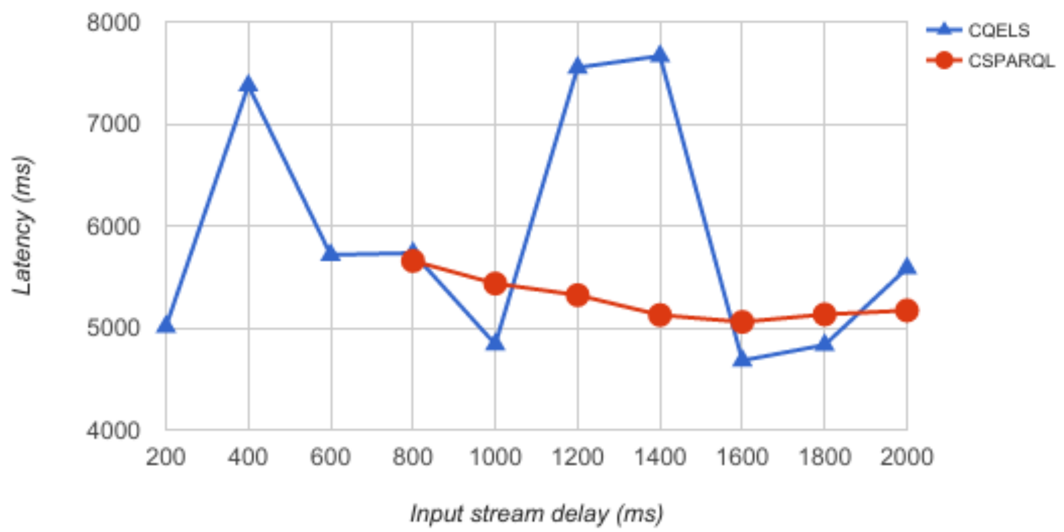

Fig. 6.7 Q1 over various input rate with large time windows and 50 concurrent queries

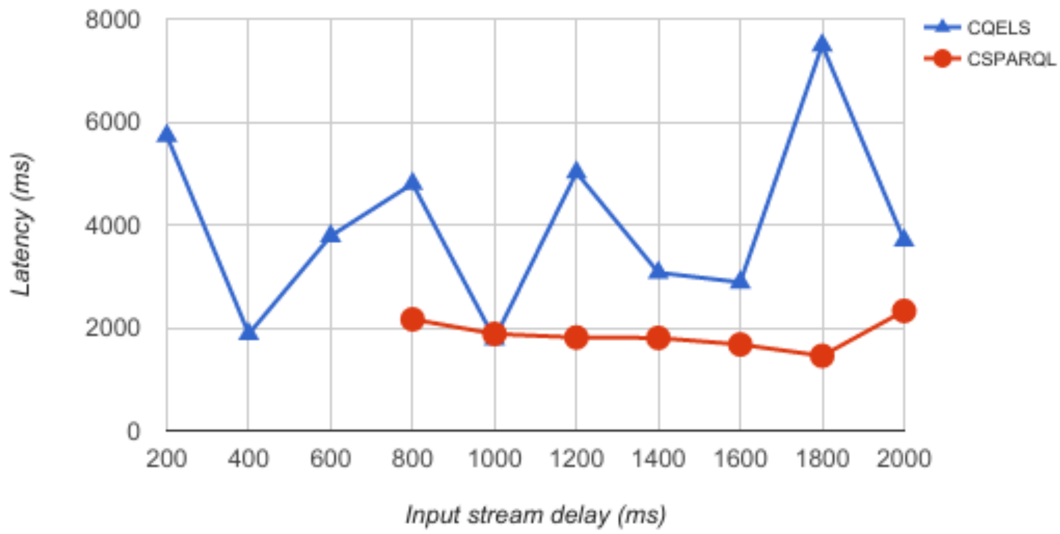

Fig. 6.8 Q1 over various input rate with small time windows and 20 concurrent queries

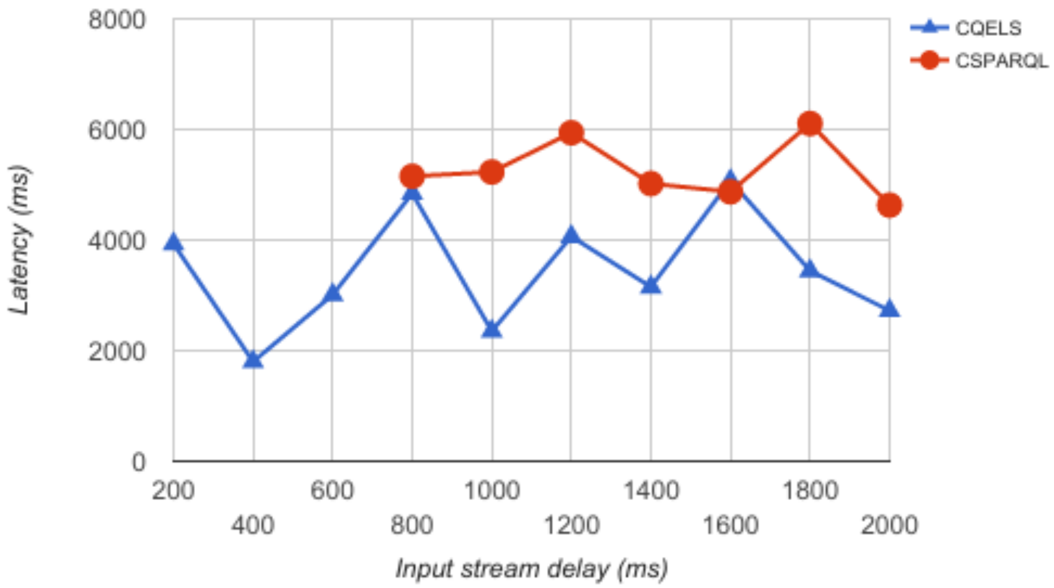

Fig. 6.9 Q1 over various input rate with small windows and 50 concurrent queries

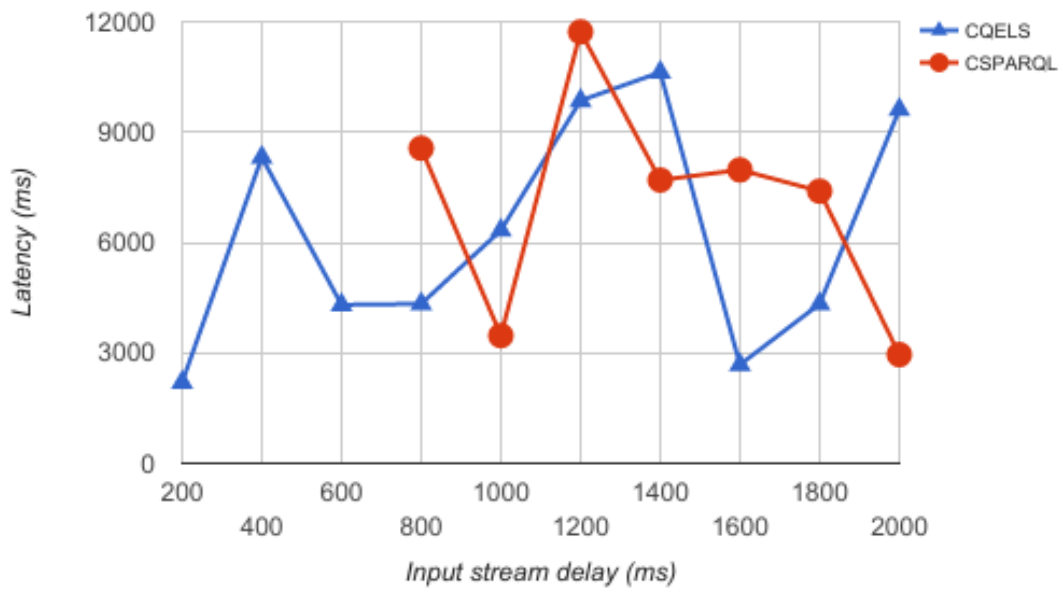

Fig. 6.10 Q1 over various input rate with small time windows and 100 concurrent queries

6.3.2 Memory Consumption

Unlike the latency memory consumption can be evaluated even when RSP engines stop responding. Hence we have evaluated memory consumption by keeping the same parameters and settings of latency with all three CityBench queries. Figure 6.11 to 6.20 depicted the memory consumption behaviour in both engines. CQELS consumed huge memory in Query1 compare to CSPARQL at every experiment. The reason is Query1 has two streams in the query and CQELS consumes huge memory to deal with multiple streams. However, CSPARQL consumes significantly less memory than CQELS in Query1. Memory consumption in CQELS is not higher than CSPARQL with Query2 till the 20 concurrent queries as depicted in Figure 6.11 and Figure 6.12. Consequently, CQELS consume more memory than CSPARQL when the number of concurrent queries increases. CQELS and CSPARQL both consumed more memory when an input stream rate is higher i.e. input stream delay is lower. CSPARQL did not consume less memory than CQELS with a smaller window. Additionally, with default window, both engines consumed slightly more memory than the larger window. Our adaptive layer can monitor the memory consumption and also controls the input rate whenever required. Though the memory consumption differences

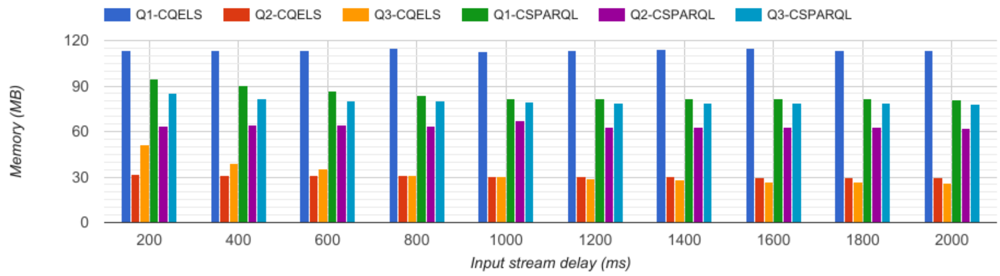

Fig. 6.11 Memory consumption at various input rate

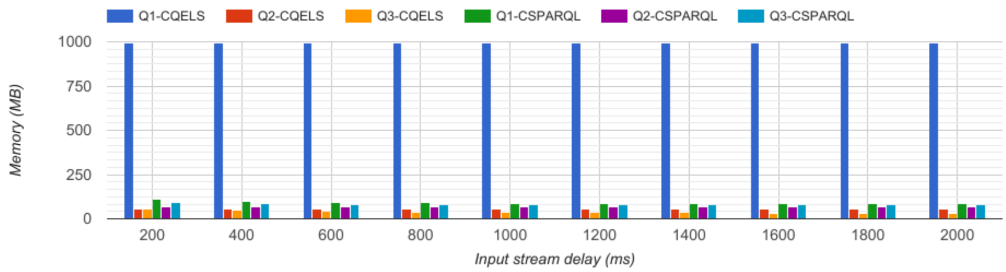

Fig. 6.12 Memory consumption at various input rate with 20 concurrent queries

at engine level may look smaller with few queries, in reality, this engine should be capable enough to process a huge number of queries, especially for SIoT objects.

6.3.3 Breaking points

We observe different critical points (breaking points) where given engines stop responding and no response from the engine was received.

Both CQELS and CSPARQL stopped responding for Query2 and Query3 at certain points as shown in Figure 6.21 and Figure 6.22. This evaluation is also helpful to figure

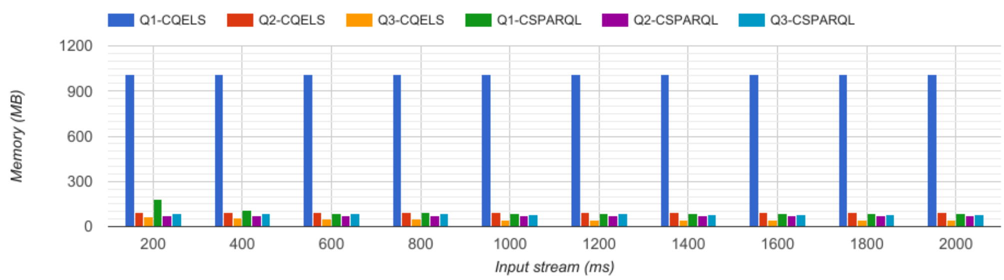

Fig. 6.13 Memory consumption at various input rate with 50 concurrent queries

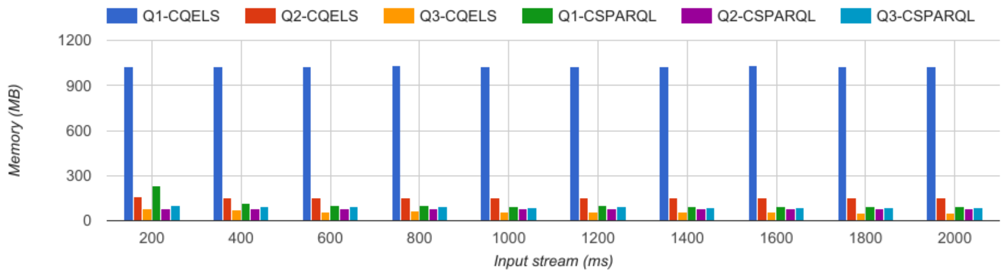

Fig. 6.14 Memory consumption at various input rate with 100 concurrent queries

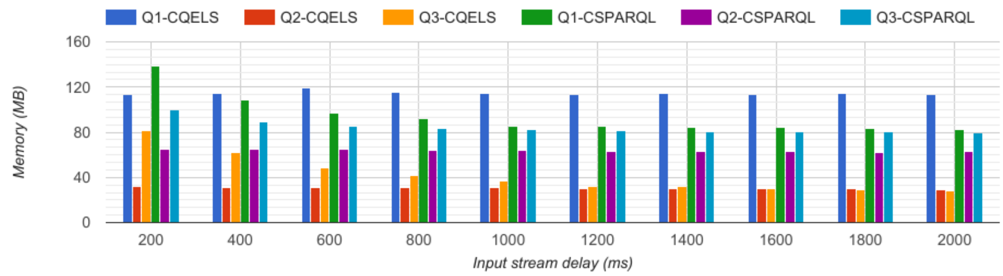

Fig. 6.15 Memory consumption at various input rate with large time windows

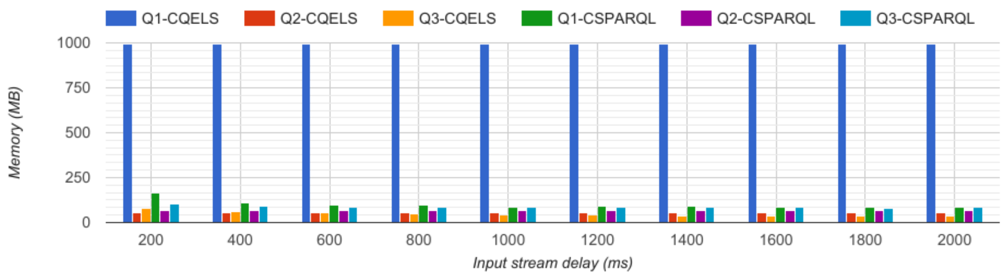

Fig. 6.16 Memory consumption at various input rate with large time windows and 20 concurrent queries

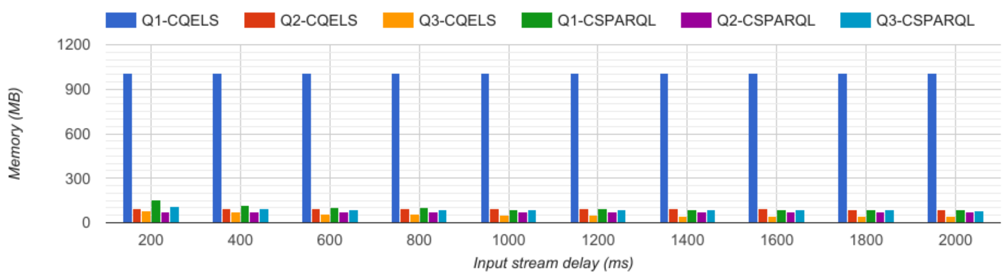

Fig. 6.17 Memory consumption at various input rate with large time windows and 50 concurrent queries

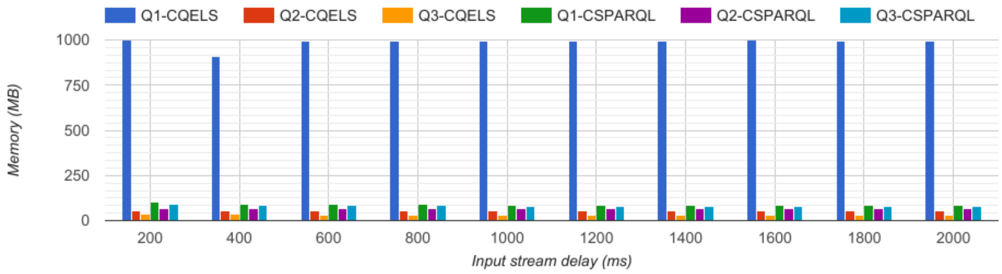

Fig. 6.18 Memory consumption at various input rate with small time windows and 20 concurrent queries

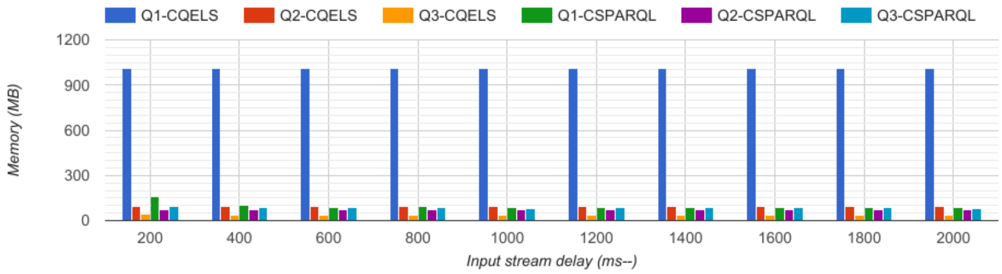

Fig. 6.19 Memory consumption at various input rate with small time windows and 50 concurrent queries

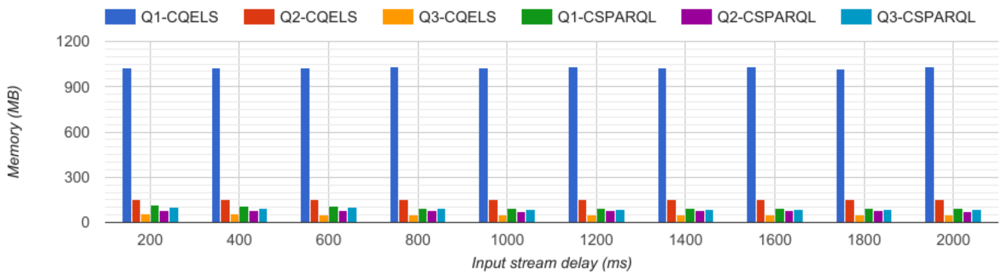

Fig. 6.20 Memory consumption at various input rate with small time windows and 100 concurrent queries

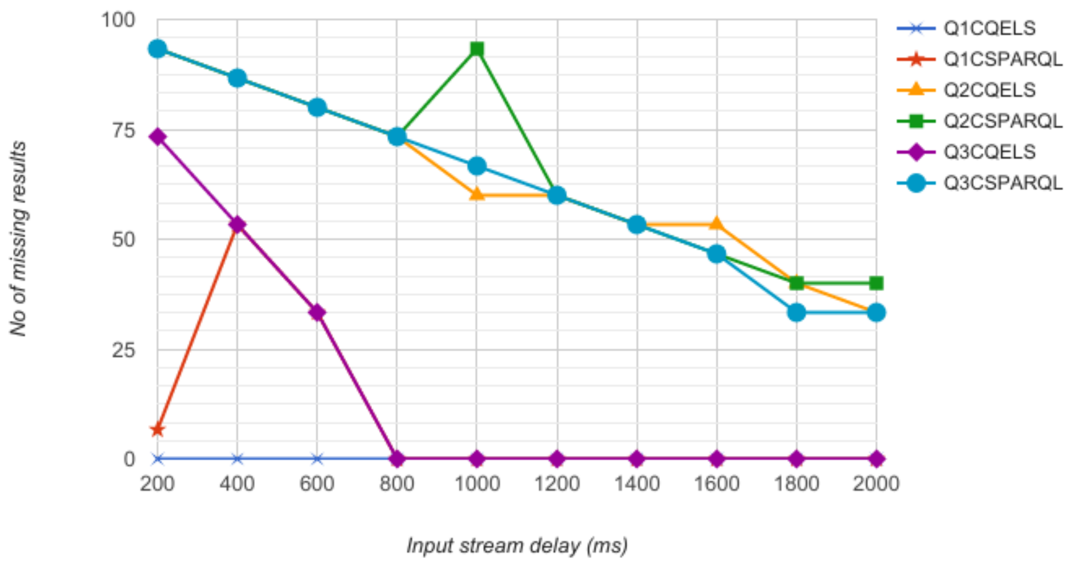

Fig. 6.21 Percentage chart of No results received

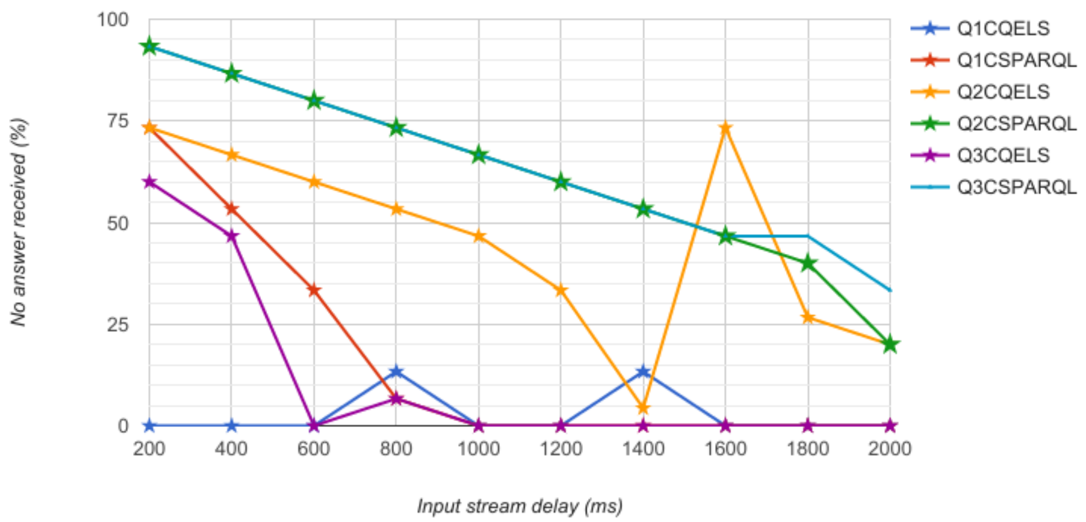

Fig. 6.22 Percentage chart of No results received with 100 concurrent queries

out at what settings RSP engines stopped responding. Especially when the input rate is high i.e. input stream delay is very low like 200 ms to 800 ms that time both of the engines stopped responding and the trend is, misses fewer results towards higher input stream delay. However, CQELS never stopped responding on Query 1 and CSPARQL never responded on 200ms, 400ms and 600ms input stream delay for all three queries.

6.4 Adaptive layer for RSP

SIoT-MF may require the different combination of RSP capabilities i.e. design time & runtime features. The present status of RSP engines (free resources) is crucial to select a suitable RSP engine. We consider present status, window size, concurrent queries and input stream rate as run-time features of RSP engines. We also intend to control the input stream rate and output results rate, where it requires to keep the resource consumption as low as possible of the RSP engines. Additionally, engine level fail-over can also be handled by the adaptive layer through monitoring underlying engines.

6.4.1 Adaptive Layer

Adaptive RSP approach consists of two main segments i.e. adaptive layer and RSP engines. Adaptive layer passes the input stream and query to an RSP engine and an RSP engine passes output results to an application through the adaptive layer. We detail all components of the proposed approach in the next this section. Adaptive layer plays the role of a middleware between application and RSP engines. Applications send a request to adaptive layer and wait for the results. In any individual RSP engine, a request directly goes to the RSP engine and starts execution without considering the input stream rate, RSP engine's processing status and most importantly application's requirements and RSP capabilities. As a result, the entire system may get halted whenever an error occurs. However, in the adaptive approach, the adaptive layer analyses query before registering to an RSP engine by considering the status of underlying components and finds the best match among RSP engines. The adaptive layer comprises six components which are described below,

Negotiator this component receives the application request and divides it into three different segments i.e. controlling input and output rate, select best match RSP engine and query handler. Firstly, it checks the User ID and Query ID inside the received request, then extracts the expected result rate of the application. Negotiator also collects the status of

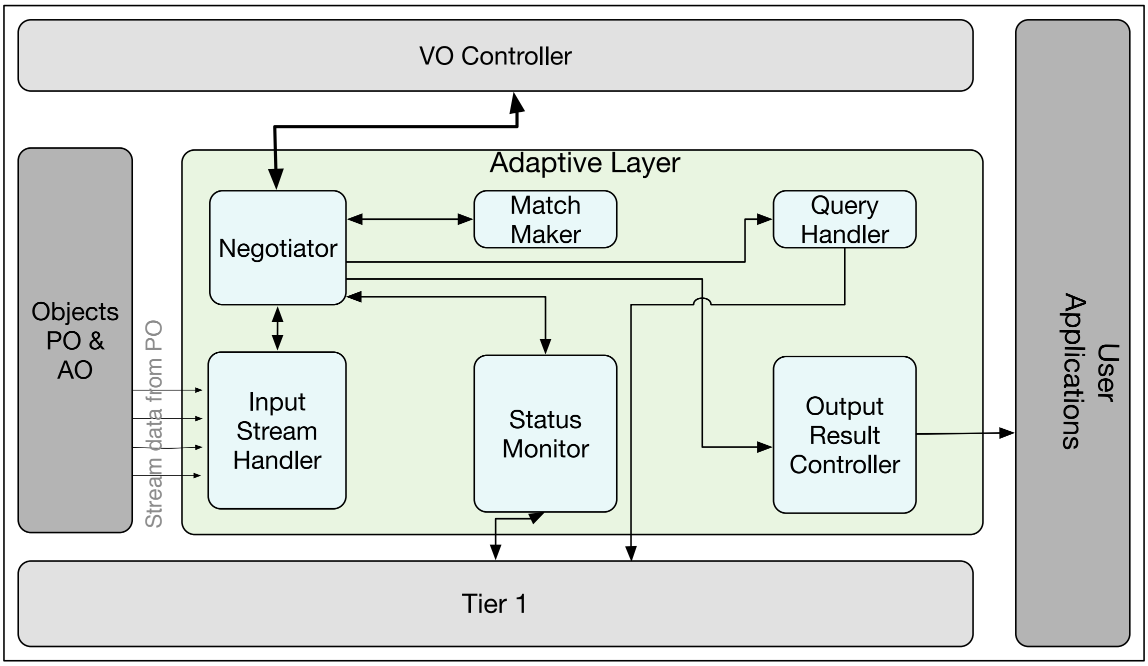

Fig. 6.23 Adaptive stream processing in SIoT-MF

RSP engines and the input stream rates. Input stream rate is decided after getting current status from input stream handler.

Input stream handler the main responsibility of this component is to receive RDF stream from POs and feed them to the RSP engines. It is able to calculate the stream arrival rate and select the data model i.e. point based or interval based. In the point based mode data arrives at any time but in the interval based mode data arrives at a certain interval. Input stream handler can also control the rate at which the RSP engine will be fed, another important role of this module is to send input stream rate to the adaptive layer of a particular stream, and its own processing capability. This component sets stream pushing rate to RSP engine according to the adaptive layer's instruction.

Status monitor collects resource consumption of available RSP engines periodically and when asked by the negotiator. The reason behind the periodic information collection is to keep monitoring the status. There is a resources consumption threshold, and if any RSP engine goes beyond the threshold then status monitor immediately reports to the negotiator. This process helps the negotiator to send the notification to the user applications.

Matchmaker this module is the key component for finding a suitable engine. After getting all the requirements, input stream handler, status monitor asks the matchmaker through negotiator to select the suitable engine.

Query handler currently, all the RSP engines have different query languages, though they are extended from SPARQL they differ from each other. W3C RSP Working Group [3] is working to form a common query language for RSP and it will be a great complement to the adaptive RSP. However, at this moment we have to handle the different query languages. This module generates and transforms query for both engines when requires.

Output result controller this component controls the output stream rate according to the instruction from negotiator. If this module receives results faster than the application expects then it may discard some results, but it is always better to control the execution than controlling the output, that is why we introduce an input stream handler. If there is no input stream then there is no unnecessary query execution. However, to handle a situation where the same input stream is required by more than one applications with various expected result rate, we introduce output controller to control the output result rate according to application's expectation. Controlling the input and output rate is also important to maintain a balance among all the components. By doing this we are able to avoid the unnecessary processing in the RSP engines.

6.4.2 Engine Selection

Selecting the suitable RSP engine based on requirements is a complex calculation as it requires both *design time features* and *run time features*. we consider *design time features*, because of their impact on output results [208, 61, 116] and RSP engine's status (free resources) as run time features. We use the MCDM technique in Chapter 4 for selecting suitable VO from a pool of VO. We use the same technique here again to select a suitable RSP engine. We use the five design time and runtime features. But other features can also be included without any fundamental changes. The features are numbered from 1 to 5. 1 = Execution 2 = Output operator 3 = Empty relation notification 4 = Input rate 5 = Memory consumptions

Given a query Qj in the adaptive layer, there is a set of candidate RSP engines $Ej = E1j, E2j, ...Enj$, that can be used to register the query once and execute continuously

[3] https://www.w3.org/community/rsp/

for continuous results. By merging all the design time and runtime features of all the RSP engines a matrix $F = (Fi, j; 1 <= i <= n, 1 <= j <= 5)$ is built, in which each row Fj corresponds to an RSP engine Eij while each column corresponds to a feature dimension. Some features are positive i.e. choosing them to increase user satisfaction and also helps to occupy system resources without overloading. On the other hand, some features are negative i.e. the higher the value, the lower the quality. Two more features we did not consider in matchmaking window size and execution time. The reason behind that user decide the window size and increasing or decreasing the size may have positive or negative effects that are showed in the evaluation section and the execution time is unknown before executing the query.

- Positive features: Execution strategy, Output Operator, Empty relation notification
- Negative Features: Input rate, Memory consumption.

Hence positive features are aligned with equation 1 and negative features are aligned with equation 2. Design time features are,

$$V_{i,j} = \begin{cases} \frac{F_j{}^{max} - F_{i,j}}{F_j{}^{max} - F_j{}^{min}} & \text{if } F_j^{max} \neq 0 \\ 1 & \text{if } F_j^{max} = 0 \\ \end{cases} \tag{1}$$

$$V_{i,j} = \begin{cases} \frac{F_{i,j} - F_j{}^{min}}{F_j{}^{max} - F_j{}^{min}} & \text{if } F_j^{max} \neq 0 \\ 1 & \text{if } F_j^{max} = 0 \\ \end{cases} \tag{2}$$

Where F_j^{max} is the maximum value of a feature in the matrix F i.e. $F_j^{max} = Max(F_{i,j}), 1 \leq i \leq n$ while F_j^{max} is the minimal value of a feature in the matrix F i.e. $F_j^{min} = Min(F_{i,j}), 1 \leq i \leq n$ while F_j^{max}, Hence we set a matrix $V = (V_{i,j}; 1 \leq i \leq n, 1 \leq j \leq 5)$ in which each row V_j corresponds to a RSP engine E_i, j while each column corresponds to a feature dimension.

Now we calculate the score of each RSP engines using the following formula, $Score(s_i)$ $= \sum_{j=1}^{5}(V_{i,j} * W_j)$, where $W_j \epsilon [0,1]$ and $\sum_{j=1}^{5} W_j = 1$. W_j represents the weight of criterion j. Users express their preferences according to the weight of each features. Finally based on the highest score the best suitable engine is selected.

In this chapter, we describe the architecture of adaptive layer for switching between two RSP engines, in our case CQELS, CSPARQL. The main idea is to enable real-time processing in SIoT-MF as well as to get the optimum performance from the available RSP engines.

6.5 Conclusion

Adaptive real-time processing is one of the major components of SIoT-MF. RSP engines support the development of smart applications with the capability of reasoning and event correlation, but current RSP engines lack adaptivity, which is a key solution for improving RSP engines results in diverse settings. RSP engines differ on a wide range of aspects including operational semantics, input data model, execution time, throughput, target domain of applications and more. To overcome this monotonous trend we have proposed an adaptive layer that can adapt to the application and also consider the input data properties, output data properties and underlying systems. We believe that addressing data and application requirements can make a difference in the correctness of RSP query results and systems resource utilisation. In this chapter, we considered a few significant features of RSP engines for the matchmaking process. We have used MCDM technique again to select the best suitable engine. In the next chapter, we describe a smart shopping scenario as a use case of SIoT-MF.

Chapter 7

Conclusion and Future Directions

The thesis delineated to pursue the management of SIoT objects by introducing Social IoT Management Framework (SIoT-MF) and their usage, where real-time and heterogeneous data processing are enabled. The study has investigated how to manage heterogeneous socially connected IoT object and select a suitable object from a pool of candidate objects to compose them in the virtual world for creating real-world services. The state-of-the-art research in Social IoT has not produced ample models or methods for managing and selecting a suitable object within a single framework along with real-time monitoring of objects and create the virtual counterpart of the real-world objects.

The study intends to answer the following research questions:

1. How to enable a better representation of the physical object to the virtual world?
2. How to select suitable objects from a pool of objects over multiple constraints and user preferences?
3. How to monitor and track the current status of objects and update their parameter in real-time?
4. How SIoT based social recommendation can be helpful in human life?

7.1 Answer to the research questions

In this section, the answer for the research questions are summarised based on the solutions provided in this thesis.

RQ1: How to enable better representation of physical object to the virtual world? We enable the IoT object virtualization in Tier 2 of the SIoT-MF architecture. The virtualization bridges the gap between the real-world and cyber world by creating a virtual counterpart of the IoT objects in the cyber world e.g devices, sensors and actuators. Moreover, we extended the virtualization method to the human expertise level. We have introduced Abstract Object (AO) which is basically a representation of human skills or expertise in the cyber world, for example, bike repairing, car repairing, computer programming, delivery service, interpreting and any other kind of skills that might be required in the society.

We have presented a QoS ontology for those virtual objects (VO) in chapter 4 by extending several existing ontology models and also considered QoS metric for each VO to rank them when they are required to perform a task.

RQ2: How to select suitable objects from a pool of object over multiple constraints and user preference? Selecting the suitable objects from a pool of candidate objects to execute a task according to the user preferences is handled by two selection techniques, Multiple Constraint Decision Making (MCDM) and Integer Programming (IP). These two techniques are well known in different sectors for solving the selection problem like Web service composition, Economics, Travel planning systems and more.

We have aligned our problems with this two techniques. MCDM is suitable when a task requires only one object and no composition is required. However, IP is required when more than one object is needed to execute a task. The objects are ranked according to the QoS score and the user preferences for each QoS criteria is considered to select a suitable object. The selection techniques are explained and evaluated in chapter 5.

RQ3: How to monitor and track the current status of objects and update their parameter in real-time? IoT objects generate data continuously and the more recent data are more relevant for the user applications normally. Moreover, any change in the object profile also needs to be monitored at real-time for the sake of continuity of an ongoing task and for the user satisfaction as well. We have considered a semantic relation among objects and semantic data as we have explained in Chapter 4 and Chapter 5. As a result to handle the semantic RDF data in real-time we have integrated existing RDF stream processing engines CQELS and CSPARQL in the SIoT-MF.

However, there are some differences in the existing engines and we have proposed an adaptive layer to make the system adaptive for the VO and CVO requirements. Chapter 5 presents the real-time processing facility of the SIoT-MF and the adaptivity at real-time RDF stream processing.

RQ4: How SIoT based social recommendation can be helpful in human life? The proposed SIoT-MF aims to enable a cognitive contributory society where physical objects will be socially connected and human will be able to share their expertise to create a service by combining the physical objects in real-time. We have presented a smart shopping scenario to show the usage of SIoT-MF. In the use case example, refrigerators are connected with each other in a society and it can sense when an item is required. As a result, the refrigerator collects recommendations from other socially connected refrigerators to select an item.

Finally, the item is ordered in either online or onsite shop. For the onsite shop, SIoT-MF enables the delivery service through the socially connected human. A smart shopping application is considered based on SIoT-MF to order and reminder facilities to the SIoT objects.

7.2 Main Contributions

The core contributions of this thesis are the provision of the integrated solution that we called SIoT-MF, which enables cognitive contributory societies and bridge the gap between human and physical objects based on SIoT, virtualization and real-time processing. We categorize the contributions in the following;

Enabling social relation among human and objects. The SIoT-MF enabled social relationship among physical objects and human. Interoperability and automation ar achieved by integrating physical objects and human skills in day-to-day life. The architecture of SIoT-MF in chapter 4 shows the integration of different state-of-the-art technologies to bridge the gap between human and physical objects.

Ontology based VO Modelling. In this thesis, we have proposed two ontologies in chapter 4 i.e., SIoT ontology and QoS ontology based on several existing ontologies. The SIoT ontology includes the relationship among VO, CVO and their owners' relationship. QoS ontology derived from the SIoT ontologies to describe the objects capability with functional properties.

Selection mechanism. Selecting a suitable object from a pool of candidate objects according to user preferences by considering the objects QoS ranking is a complex task. Constraint-based two mechanisms namely, MCDM and IP are adopted to solve this problem. The evaluation in Chapter 5 shows that MCDM consumes fewer resources compare to IP. However, IP is suitable when more than one VO is required to execute a task.

Real-time monitoring. SIoT-MF enables real-time monitoring of objects to keep the freshness of the data. We adopted RDF stream processing engines i.e., CQELS and CSPARQL in the SIoT-MF. Moreover, we have presented an adaptive layer to handle the differences among these two engines by considering the application requirements and selecting the most suitable engine on the fly.

Utilization of socially connected objects. We have presented a smart shopping scenario based on SIoT-MF. Where human skills and object capabilities are combined to solve a real-world societal challenge. An object can detect and collects recommendations with the help of SIoT-MF to select an item and then a suitable shop to buy the item. Eventually, a smart shopping application helps to buy and deliver through the socially connected human. Additionally, we have presented a SIoT based shopping QoS ontology in the scenario.

7.3 Limitations of the Study

We discuss the limitations of this study in the following:

- There is no social network for the physical object according to our knowledge till date. To realize SIoT-MF a real social network is required that accept physical objects.
- A distributed processing facilities are required to avoid resource limitations for SIoT-MF.
- The dataset we have used for evaluating the different object selection mechanism is not a real dataset. Instead, we have used a synthetic dataset according to the ontology model.
- Few more use case scenario along with full implementation is required rather than a prototype implementation.

- Energy management is another aspect to evaluate a system's performance which is not included in this thesis.
- This thesis did not consider security, privacy and trust which is a sensitive issue for the real world success of SIoT. It is not possible to gain acceptance without securing the entire system. As a result, it is a great limitation of this study that should be considered in the future study.

7.4 Future Work

There are several challenges towards cognitive contributory societies through SIoT-MF. We intend to address those challenges in our future work. In this section, we present the future work to handle those challenges.

Object social network. Presently there is no online social network platform that facilitates physical object profile. To enable cognitive societies the essential part is to create a platform that provides the opportunity to physical objects to form an online social network along with human.

Resource management. Currently, the SIoT-MF is a centralised system and computing resource consumption and energy consumption are great obstacles to provide seamless service. Moreover, a great number of mobile devices and tiny SIoT nodes are part of SIoT-MF which run on limited processing capabilities and tiny batteries. One solution could be to outsource all the task of SIoT-MF that demand high processing capabilities to the cloud. This approach will save energy in local devices as long energy sourcing technologies for tiny devices do not provide any better solution.

Trust, privacy and security. This is the key for SIoT-MF to achieve success in the real world. The main point of making a friend in human societies is trust and it is the same in the SIoT as well. Firstly, a system should be secured and Secondly, the system must be able to maintain the privacy of the users to be trusted by the users. the good point is SIoT-MF may adopt existing methods and technologies to ensure data confidentiality and user privacy from similar platforms. However, it is not very simple and straightforward adoption because of the heterogeneity and dynamicity of the SIoT nodes. Hence, a significant amount of research needs to be done to consider SIoT-MF a secured and trusted framework.

Bussiness case. SIoT has not penetrated in the industry yet, therefore, new product model and business case scenario are required to make SIoT more visible to the industry. As a result, we plan to publish the smart shopping scenario and application online eventually.

Standardization. We believe that community involvement is essential to finalize standards and technologies of SIoT. We intend to submit our proposed SIoT-MF architecture in the IoT standardization body.

References

[1] (2012). icore, “d3.1: Virtual object requirements and dependencies. Technical report.

[2] (2013a). Behaviorally segmenting the internet of things (iot).

[3] (2013b). icore, “d3.2: Real object awareness and association. Technical report.

[4] (2013c). Requirements for the industrial internet of things.

[5] (2014). The industrial internet of things.

[6] Ahmed, S. H., Kim, G., and Kim, D. (2013). Cyber physical system: Architecture, applications and research challenges. In *Wireless Days (WD), 2013 IFIP*, pages 1–5. IEEE.

[7] Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). A survey on sensor networks. *IEEE communications magazine*, 40(8):102–114.

[8] Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys & Tutorials*, 17(4):2347–2376.

[9] Ali, M. I., Gao, F., and Mileo, A. (2015). Citybench: A configurable benchmark to evaluate rsp engines using smart city datasets. In *The Semantic Web-ISWC 2015*. Springer.

[10] ALL, T. (2013). Collaborative open market to place objects at your service. Technical report, Tech. Rep.

[11] Aloi, G., Caliciuri, G., Fortino, G., Gravina, R., Pace, P., Russo, W., and Savaglio, C. (2017). Enabling iot interoperability through opportunistic smartphone-based mobile gateways. *Journal of Network and Computer Applications*, 81:74–84.

[12] An, J., Gui, X., Zhang, W., Jiang, J., and Yang, J. (2013). Research on social relations cognitive model of mobile nodes in internet of things. *Journal of Network and Computer Applications*, 36(2):799–810.

[13] Andrea, I., Chrysostomou, C., and Hadjichristofi, G. (2015). Internet of things: Security vulnerabilities and challenges. In *Computers and Communication (ISCC), 2015 IEEE Symposium on*, pages 180–187. IEEE.

[14] Anicic, D., Fodor, P., Rudolph, S., and Stojanovic, N. (2011). EP-SPARQL: a unified language for event processing and stream reasoning. In *Proc. of the 20th WWW Conference*, pages 635–644. ACM.

[15] Arasu, A., Babu, S., and Widom, J. (2004). Cql: A language for continuous queries over streams and relations. In *Database Programming Languages*, pages 1–19. Springer.

[16] Atzori, L., Iera, A., and Morabito, G. (2010a). The internet of things: A survey. *Computer networks*, 54(15):2787–2805.

[17] Atzori, L., Iera, A., and Morabito, G. (2010b). The internet of things: A survey. *Computer networks*, 54(15):2787–2805.

[18] Atzori, L., Iera, A., and Morabito, G. (2011a). Making things socialize in the internet—does it help our lives? In *Kaleidoscope 2011: The Fully Networked Human?-Innovations for Future Networks and Services (K-2011), Proceedings of ITU*, pages 1–8. Citeseer.

[19] Atzori, L., Iera, A., and Morabito, G. (2011b). Siot: Giving a social structure to the internet of things. *IEEE communications letters*, 15(11):1193–1195.

[20] Atzori, L., Iera, A., and Morabito, G. (2014). From" smart objects" to" social objects": The next evolutionary step of the internet of things. *IEEE Communications Magazine*, 52(1):97–105.

[21] Atzori, L., Iera, A., Morabito, G., and Nitti, M. (2012a). The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. *Computer networks*, 56(16):3594–3608.

[22] Atzori, L., Iera, A., Morabito, G., and Nitti, M. (2012b). The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. *Computer networks*, 56(16):3594–3608.

[23] Atzori, L., Iera, A., Morabito, G., and Nitti, M. (2012c). The social internet of things (siot)–when social networks meet the internet of things: Concept, architecture and network characterization. *Computer networks*, 56(16):3594–3608.

[24] Bai, J. (2013). Feasibility analysis of big log data real time search based on hbase and elasticsearch. In *Natural Computation (ICNC), 2013 Ninth International Conference on*, pages 1166–1170. IEEE.

[25] Bakillah, M., Liang, S. H., Zipf, A., and Mostafavi, M. A. (2013). A dynamic and context-aware semantic mediation service for discovering and fusion of heterogeneous sensor data. *Journal of Spatial Information Science*, 2013(6):155–185.

[26] Bandyopadhyay, D. and Sen, J. (2011). Internet of things: Applications and challenges in technology and standardization. *Wireless Personal Communications*, 58(1):49–69.

[27] Bandyopadhyay, S., Sengupta, M., Maiti, S., and Dutta, S. (2011). Role of middleware for internet of things: A study. *International Journal of Computer Science and Engineering Survey*, 2(3):94–105.

[28] Bao, F., Chen, R., and Guo, J. (2013). Scalable, adaptive and survivable trust management for community of interest based internet of things systems. In *Autonomous Decentralized Systems (ISADS), 2013 IEEE Eleventh International Symposium on*, pages 1–7. IEEE.

[29] Baqer, M. (2010). Enabling collaboration and coordination of wireless sensor networks via social networks. In *Distributed Computing in Sensor Systems Workshops (DCOSSW), 2010 6th IEEE International Conference on*, pages 1–2. IEEE.

[30] Baqer, M. and Kamal, A. (2009). S-sensors: Integrating physical world inputs with social networks using wireless sensor networks. In *Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2009 5th International Conference on*, pages 213–218. IEEE.

[31] Barbieri, D. F., Braga, D., Ceri, S., Della Valle, E., and Grossniklaus, M. (2009). C-sparql: Sparql for continuous querying. In *Proc. of the 18th*, World Wide Web Conf., pages 1061–1062, New York, NY, USA. ACM.

[32] Baronti, P., Pillai, P., Chook, V. W., Chessa, S., Gotta, A., and Hu, Y. F. (2007). Wireless sensor networks: A survey on the state of the art and the 802.15. 4 and zigbee standards. *Computer communications*, 30(7):1655–1695.

[33] Barrett, R. and Maglio, P. P. (1998). Informative things: How to attach information to the real world. In *Proceedings of the 11th annual ACM symposium on User interface software and technology*, pages 81–88. ACM.

[34] Bélissent, J. et al. (2010). Getting clever about smart cities: New opportunities require new business models. *Cambridge, Massachusetts, USA*.

[35] Berners-Lee, T., Hendler, J., and Lassila, O. (2001). The semantic web. *Scientific american*, 284(5):34–43.

[36] Blackstock, M., Lea, R., and Friday, A. (2011). Uniting online social networks with places and things. In *Proceedings of the Second International Workshop on Web of Things*, page 5. ACM.

[37] Bolles, A., Grawunder, M., and Jacobi, J. (2008). Streaming sparql extending sparql to process data streams. In *Proc. of the 5th ESWC: Research and Applications*, ESWC'08, pages 448–462, Berlin, Heidelberg. Springer-Verlag.

[38] Borgia, E. (2014). The internet of things vision: Key features, applications and open issues. *Computer Communications*, 54:1–31.

[39] Bormann, C., Castellani, A. P., and Shelby, Z. (2012). Coap: An application protocol for billions of tiny internet nodes. *IEEE Internet Computing*, 16(2):62–67.

[40] Botta, A., De Donato, W., Persico, V., and Pescapé, A. (2014). On the integration of cloud computing and internet of things. In *Future internet of things and cloud (FiCloud), 2014 international conference on*, pages 23–30. IEEE.

[41] Boyd, D. M. and Ellison, N. B. (2007). Social network sites: Definition, history, and scholarship. *Journal of computer-mediated Communication*, 13(1):210–230.

[42] Calbimonte, J.-P., Jeung, H., Corcho, O., and Aberer, K. (2012). Enabling query technologies for the semantic sensor web. *International Journal On Semantic Web and Information Systems (IJSWIS)*, 8(1):43–63.

[43] Cardenas, A. A., Amin, S., and Sastry, S. (2008). Secure control: Towards survivable cyber-physical systems. In *Distributed Computing Systems Workshops, 2008. ICDCS'08. 28th International Conference on*, pages 495–500. IEEE.

[44] Cardoso, J., Sheth, A., Miller, J., Arnold, J., and Kochut, K. (2004). Quality of service for workflows and web service processes. *Web Semantics: Science, Services and Agents on the World Wide Web*, 1(3):281–308.

[45] Casadei, R., Fortino, G., Pianini, D., Russo, W., Savaglio, C., and Viroli, M. (2019). Modelling and simulation of opportunistic iot services with aggregate computing. *Future Generation Computer Systems*, 91:252–262.

[46] Ceipidor, U. B., Medaglia, C. M., Volpi, V., Moroni, A., Sposato, S., and Tamburrano, M. (2011). Design and development of a social shopping experience in the iot domain: The shoplovers solution. In *Software, Telecommunications and Computer Networks (SoftCOM), 2011 19th International Conference on*, pages 1–5. IEEE.

[47] Chaqfeh, M. A. and Mohamed, N. (2012). Challenges in middleware solutions for the internet of things. In *Collaboration Technologies and Systems (CTS), 2012 International Conference on*, pages 21–26. IEEE.

[48] Chen, F., Xiang, T., Fu, X., and Yu, W. (2016a). User differentiated verifiable file search on the cloud. *IEEE Transactions on Services Computing*.

[49] Chen, Z., Xu, G., Mahalingam, V., Ge, L., Nguyen, J., Yu, W., and Lu, C. (2016b). A cloud computing based network monitoring and threat detection system for critical infrastructures. *Big Data Research*, 3:10–23.

[50] Chowdhury, N. M. K. and Boutaba, R. (2009). Network virtualization: state of the art and research challenges. *IEEE Communications magazine*, 47(7).

[51] Chowdhury, N. M. K. and Boutaba, R. (2010). A survey of network virtualization. *Computer Networks*, 54(5):862–876.

[52] Ćika, D., Draganić, M., and Šipuš, Z. (2012). Active wireless sensor with radio frequency identification chip. In *MIPRO, 2012 Proceedings of the 35th International Convention*, pages 727–732. IEEE.

[53] Ciortea, A., Boissier, O., Zimmermann, A., and Florea, A. M. (2013). Reconsidering the social web of things: position paper. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 1535–1544. ACM.

[54] Committee, L. S. et al. (2002). Ieee standard for local and metropolitan area networks: Overview and architecture (en línea). *New York-NY-USA. The Institute of Electrical and Electronics Engineers Inc.*

[55] Compton, M., Barnaghi, P., Bermudez, L., GarcíA-Castro, R., Corcho, O., Cox, S., Graybeal, J., Hauswirth, M., Henson, C., Herzog, A., et al. (2012). The ssn ontology of the w3c semantic sensor network incubator group. *Web semantics: science, services and agents on the World Wide Web*, 17:25–32.

[56] Console, L., Antonelli, F., Biamino, G., Carmagnola, F., Cena, F., Chiabrando, E., Cuciti, V., Demichelis, M., Fassio, F., Franceschi, F., et al. (2013). Interacting with social networks of intelligent things and people in the world of gastronomy. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 3(1):4.

[57] Da Xu, L. (2011). Enterprise systems: State-of-the-art and future trends. *IEEE Trans. Industrial Informatics*, 7(4):630–640.

[58] Da Xu, L., He, W., and Li, S. (2014a). Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, 10(4):2233–2243.

[59] Da Xu, L., He, W., and Li, S. (2014b). Internet of things in industries: A survey. *IEEE Transactions on industrial informatics*, 10(4):2233–2243.

[60] Della Valle, E., Ceri, S., Van Harmelen, F., and Fensel, D. (2009). It's a streaming world! reasoning upon rapidly changing information. *IEEE Intelligent Systems*, (6):83–89.

[61] Dell'Aglio, D., Calbimonte, J.-P., Balduini, M., Corcho, O., and Valle, E. D. (2013). On correctness in rdf stream processor benchmarking. In *ISWC (2)*, volume 8219 of *LNCS*, pages 326–342. Springer.

[62] Di Modica, G., Tomarchio, O., and Vita, L. (2011). Resource and service discovery in soas: A p2p oriented semantic approach. *International Journal of Applied Mathematics and Computer Science*, 21(2):285–294.

[63] Dominique, G. (2011). A web of things application architecture-integrating the real-world into the web. *Zurich, Diss. ETH*, (19891):10–12.

[64] Dong, S., D. W. (2009). A qos driven web service composition method based on esga (elitist selection genetic algorithm) with an improved initial ppulation selection strategy. *International Journal of Distributed Sensor Networks.*

[65] Doukas, C. and Antonelli, F. (2013). Compose: Building smart & context-aware mobile applications utilizing iot technologies. In *Global Information Infrastructure Symposium, 2013*, pages 1–6. IEEE.

[66] Du, X., Xiao, Y., Guizani, M., and Chen, H.-H. (2007). An effective key management scheme for heterogeneous sensor networks. *Ad Hoc Networks*, 5(1):24–34.

[67] Duce, H. (2008). Internet of things in 2020.

[68] Ellis, S. R. (1994). What are virtual environments? *IEEE Computer Graphics and Applications*, 14(1):17–22.

[69] Fandel, Günter, G. T. H. T. (1997). *Multiple criteria decision making*. Springer.

[70] Farris, I., Iera, A., Militano, L., and Spinella, S. C. (2014). Performance evaluation of rfid tag-based" virtual" communication channels. In *ICC*, pages 2897–2902.

[71] Finkenzeller, K. (2010). *RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication*. John Wiley & Sons.

[72] Fortino, G., Gravina, R., Russo, W., and Savaglio, C. (2017a). Modeling and simulating internet-of-things systems: a hybrid agent-oriented approach. *Computing in Science & Engineering*, 19(5):68–76.

[73] Fortino, G., Guerrieri, A., Russo, W., and Savaglio, C. (2014). Middlewares for smart objects and smart environments: overview and comparison. In *Internet of Things Based on Smart Objects*, pages 1–27. Springer.

[74] Fortino, G., Guerrieri, A., Russo, W., and Savaglio, C. (2015). Towards a development methodology for smart object-oriented iot systems: a metamodel approach. In *Systems, Man, and Cybernetics (SMC), 2015 IEEE International Conference on*, pages 1297–1302. IEEE.

[75] Fortino, G., Russo, W., Savaglio, C., Shen, W., and Zhou, M. (2017b). Agent-oriented cooperative smart objects: From iot system design to implementation. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, (99):1–18.

[76] Fortino, G. and Trunfio, P. (2014). *Internet of things based on smart objects: Technology, middleware and applications*. Springer.

[77] Foteinos, V., Kelaidonis, D., Poulios, G., Vlacheas, P., Stavroulaki, V., and Demestichas, P. (2013). Cognitive management for the internet of things: A framework for enabling autonomous applications. *IEEE vehicular technology magazine*, 8(4):90–99.

[78] Fu, C., Peng, C., Liu, X.-Y., Yang, L. T., Yang, J., and Han, L. (2018). Search engine: The social relationship driving power of internet of things. *Future Generation Computer Systems*.

[79] Fu, X., Ling, Z., Yu, W., and Luo, J. (2010a). Cyber crime scene investigations (c^2si) through cloud computing. In *Distributed Computing Systems Workshops (ICDCSW), 2010 IEEE 30th International Conference on*, pages 26–31. IEEE.

[80] Fu, X., Ling, Z., Yu, W., and Luo, J. (2010b). Cyber crime scene investigations (c^2si) through cloud computing. In *Distributed Computing Systems Workshops (ICDCSW), 2010 IEEE 30th International Conference on*, pages 26–31. IEEE.

[81] G. Canfora, M. D. P. (2004). A lightweight approach for qos–aware service composition. *Proceedings of the 2nd International Conference on Service Oriented Computing.*

[82] G. Canfora, M. Di Penta, R. E. e. a. (2005). An approach for qos-aware service composition based on genetic algorithms. *Genetic and Evolutionary Computation Conference (GECCO).*

[83] Gama, K., Touseau, L., and Donsez, D. (2012). Combining heterogeneous service technologies for building an internet of things middleware. *Computer Communications*, 35(4):405–417.

[84] Gan, G., Lu, Z., and Jiang, J. (2011). Internet of things security analysis. In *Internet Technology and Applications (iTAP), 2011 International Conference on*, pages 1–4. IEEE.

[85] Gao, W., Nguyen, J., Yu, W., Lu, C., and Ku, D. (2016). Assessing performance of constrained application protocol (coap) in manet using emulation. In *Proceedings of the International Conference on Research in Adaptive and Convergent Systems*, pages 103–108. ACM.

[86] Godfrey, R., Ingham, D., and Schloming, R. (2012). Oasis advanced message queuing protocol (amqp) version 1.0.

[87] Goldberg, R. P. (1973). Architecture of virtual machines. In *Proceedings of the workshop on virtual computer systems*, pages 74–112. ACM.

[88] Goldberg, R. P. (1974). Survey of virtual machine research. *Computer*, (6):34–45.

[89] Gomez, C. and Paradells, J. (2010). Wireless home automation networks: A survey of architectures and technologies. *IEEE Communications Magazine*, 48(6).

[90] Gómez-Goiri, A. and López-de Ipiña, D. (2010). A triple space-based semantic distributed middleware for internet of things. In *International Conference on Web Engineering*, pages 447–458. Springer.

[91] Gorbach, G., Polsonetti, C., and Chatha, A. (2014). Planning for the industrial internet of things.

[92] Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future generation computer systems*, 29(7):1645–1660.

[93] Guinard, D., Fischer, M., and Trifa, V. (2010a). Sharing using social networks in a composable web of things. In *PerCom Workshops*, pages 702–707.

[94] Guinard, D., Trifa, V., Karnouskos, S., Spiess, P., and Savio, D. (2010b). Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services. *IEEE transactions on Services Computing*, (3):223–235.

[95] Guo, B., Zhang, D., and Wang, Z. (2011). Living with internet of things: The emergence of embedded intelligence. In *2011 IEEE International Conferences on Internet of Things, and Cyber, Physical and Social Computing*, pages 297–304. IEEE.

[96] Hadim, S. and Mohamed, N. (2006). Middleware: Middleware challenges and approaches for wireless sensor networks. *IEEE distributed systems online*, 7(3):1–1.

[97] Han, C., Jornet, J. M., Fadel, E., and Akyildiz, I. F. (2013). A cross-layer communication module for the internet of things. *Computer Networks*, 57(3):622–633.

[98] He, X. (2010). The two-dimensional bar code application in book management. In *2010 International Conference on Web Information Systems and Mining*, pages 409–411. IEEE.

[99] Helps, R. G. and Pack, S. J. (2013). Cyber-physical system concepts for it students. In *Proceedings of the 14th annual ACM SIGITE conference on Information technology education*, pages 7–12. ACM.

[100] Hu Chunhua, Wu Min, L. G. e. a. (2007). Workflow structure approach for web service based on service creation graph. *Journal of Software*, 18(8):1870–1882.

[101] Hu Chunhua, Wu Min, L. G. e. a. (2010). Quantum genetic algorithm for web service selection. *Journal of Xidian University*, 37(1):56–61.

[102] Hussein, D., Han, S. N., Han, X., Lee, G. M., and Crespi, N. (2013). A framework for social device networking. In *Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on*, pages 356–360. IEEE.

[103] Ilie-Zudor, E., Kemény, Z., Van Blommestein, F., Monostori, L., and Van Der Meulen, A. (2011). A survey of applications and requirements of unique identification systems and rfid techniques. *Computers in Industry*, 62(3):227–252.

[104] Jara, A. J., Martinez-Julia, P., and Skarmeta, A. (2012). Light-weight multicast dns and dns-sd (lmdns-sd): Ipv6-based resource and service discovery for the web of things. In *innovative mobile and internet services in ubiquitous computing (IMIS), 2012 sixth international conference on*, pages 731–738. IEEE.

[105] Jefferson, D. R. (1985). Virtual time. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 7(3):404–425.

[106] Katasonov, A., Kaykova, O., Khriyenko, O., Nikitin, S., and Terziyan, V. Y. (2008). Smart semantic middleware for the internet of things. *Icinco-Icso*, 8:169–178.

[107] Kelaidonis, D., Giaffreda, R., Somov, A., Foteinos, V., Poulios, G., Stavroulaki, V., Vlacheas, P., Demestichas, P., Baranov, A., and Biswas, A. R. (2012). Virtualization and cognitive management of real world objects in the internet of things. In *2012 IEEE International Conference on Green Computing and Communications*, pages 187–194. IEEE.

[108] Khan, I., Belqasmi, F., Glitho, R., and Crespi, N. (2013). A multi-layer architecture for wireless sensor network virtualization. In *6th Joint IFIP Wireless and Mobile Networking Conference (WMNC'2013) 2013*, pages 1–4.

[109] Kim, A. N., Hekland, F., Petersen, S., and Doyle, P. (2008). When hart goes wireless: Understanding and implementing the wirelesshart standard. In *Emerging Technologies and Factory Automation, 2008. ETFA 2008. IEEE International Conference on*, pages 899–907. IEEE.

[110] Kluge, W., Poegel, F., Roller, H., Lange, M., Ferchland, T., Dathe, L., and Eggert, D. (2006). A fully integrated 2.4 ghz ieee 802.15. 4 compliant transceiver for zigbee applications. In *Solid-State Circuits Conference, 2006. ISSCC 2006. Digest of Technical Papers. IEEE International*, pages 1470–1479. IEEE.

[111] Komazec, S., Cerri, D., and Fensel, D. (2012). Sparkwave: continuous schema-enhanced pattern matching over RDF data streams. In *Proc. of DEBS*, pages 58–68.

[112] Kranz, M., Roalter, L., and Michahelles, F. (2010). Things that twitter: social networks and the internet of things. In *What can the Internet of Things do for the Citizen (CIoT) Workshop at The Eighth International Conference on Pervasive Computing (Pervasive 2010)*, pages 1–10.

[113] Lahiri, S. (2005). *RFID sourcebook*. IBM press.

[114] Langheinrich, M., Mattern, F., Römer, K., and Vogt, H. (2000). First steps towards an event-based infrastructure for smart things. In *Ubiquitous Computing Workshop (PACT 2000)*, page 34.

[115] Le, T. N., Yu, W., Bai, X., and Xuan, D. (2006). A dynamic geographic hash table for data-centric storage in sensor networks. In *Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE*, volume 4, pages 2168–2174. IEEE.

[116] Le-Phuoc, D., Dao-Tran, M., Pham, M., Boncz, P., Eiter, T., and Fink, M. (2012). Linked stream data processing engines: Facts and figures. *The Semantic Web–ISWC 2012*, pages 300–312.

[117] Le-Phuoc, D., Dao-Tran, M., Xavier Parreira, J., and Hauswirth, M. (2011). A native and adaptive approach for unified processing of linked streams and linked data. *The Semantic Web–ISWC 2011*, pages 370–388.

[118] Lee, G. M., Rhee, W. S., and Crespi, N. (2013). Proposal of a new work item on social and device networking.

[119] Leo, M., Battisti, F., Carli, M., and Neri, A. (2014). A federated architecture approach for internet of things security. In *Euro Med Telco Conference (EMTC), 2014*, pages 1–5. IEEE.

[120] Lequerica, I., Longaron, M. G., and Ruiz, P. M. (2010). Drive and share: efficient provisioning of social networks in vehicular scenarios. *IEEE Communications Magazine*, 48(11):90–97.

[121] Li, K. and Hudak, P. (1989). Memory coherence in shared virtual memory systems. *ACM Transactions on Computer Systems (TOCS)*, 7(4):321–359.

[122] Li, S., Da Xu, L., and Zhao, S. (2015). The internet of things: a survey. *Information Systems Frontiers*, 17(2):243–259.

[123] Li Jinzhong, Xia Jiewu, T. W. e. a. (2010). A review of qos-based web service selection algorithms. *Application Research of Computers*, pages 3622–3627.

[124] Liang-Jie Zhang, Bing Li, T. C. e. a. (2003). On demand web services-based business process composition. *IEEE International Conference on System*.

[125] Lin, J., Yang, X., Yu, W., and Fu, X. (2011). Towards effective en-route filtering against injected false data in wireless sensor networks. In *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*, pages 1–5. IEEE.

[126] Lin, J., Yu, W., Yang, X., Xu, G., and Zhao, W. (2012). On false data injection attacks against distributed energy routing in smart grid. In *Proceedings of the 2012 IEEE/ACM Third International Conference on Cyber-Physical Systems*, pages 183–192. IEEE Computer Society.

[127] Lin, J., Yu, W., Yang, X., Yang, Q., Fu, X., and Zhao, W. (2015). A novel dynamic en-route decision real-time route guidance scheme in intelligent transportation systems. In *Distributed Computing Systems (ICDCS), 2015 IEEE 35th International Conference on*, pages 61–72. IEEE.

[128] Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., and Zhao, W. (2017a). A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet of Things Journal*, 4(5):1125–1142.

[129] Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., and Zhao, W. (2017b). A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications. *IEEE Internet of Things Journal*, 4(5):1125–1142.

[130] Liu, Y. and Zhou, G. (2012). Key technologies and applications of internet of things. In *Intelligent Computation Technology and Automation (ICICTA), 2012 Fifth International Conference on*, pages 197–200. IEEE.

[131] Lohr, S. (2012). The age of big data. *New York Times*, 11(2012).

[132] Luckenbach, T., Gober, P., Arbanowski, S., Kotsopoulos, A., and Kim, K. (2005). Tinyrest-a protocol for integrating sensor networks into the internet. In *Proc. of REAL-WSN*, pages 101–105.

[133] Mahmoud, R., Yousuf, T., Aloul, F., and Zualkernan, I. (2015). Internet of things (iot) security: Current status, challenges and prospective measures. In *Internet Technology and Secured Transactions (ICITST), 2015 10th International Conference for*, pages 336–341. IEEE.

[134] Mäkitalo, N., Pääkkö, J., Raatikainen, M., Myllärniemi, V., Aaltonen, T., Leppänen, T., Männistö, T., and Mikkonen, T. (2012). Social devices: collaborative co-located interactions in a mobile cloud. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia*, page 10. ACM.

[135] Mello, L. D. and Kubota, L. T. (2002). Review of the use of biosensors as analytical tools in the food and drink industries. *Food chemistry*, 77(2):237–256.

[136] Miorandi, D., Sicari, S., De Pellegrini, F., and Chlamtac, I. (2012a). Internet of things: Vision, applications and research challenges. *Ad hoc networks*, 10(7):1497–1516.

[137] Miorandi, D., Sicari, S., De Pellegrini, F., and Chlamtac, I. (2012b). Internet of things: Vision, applications and research challenges. *Ad hoc networks*, 10(7):1497–1516.

[138] Miraz, M. H., Ali, M., Excell, P. S., and Picking, R. (2015). A review on internet of things (iot), internet of everything (ioe) and internet of nano things (iont). In *2015 Internet Technologies and Applications (ITA)*, pages 219–224. IEEE.

[139] Misra, S., Barthwal, R., and Obaidat, M. S. (2012). Community detection in an integrated internet of things and social network architecture. In *Global Communications Conference (GLOBECOM), 2012 IEEE*, pages 1647–1652. IEEE.

[140] Modukari, K., Hariri, S., Chalfoun, N. V., and Yousif, M. (2005). Autonomous middleware framework for sensor networks. *International Journal of Pervasive Computing and Communications*, 1(4):337–345.

[141] Muniz, R., Junco, L., and Otero, A. (1999). A robust software barcode reader using the hough transform. In *Information Intelligence and Systems, 1999. Proceedings. 1999 International Conference on*, pages 313–319. IEEE.

[142] Nakamura, J. (2016). *Image sensors and signal processing for digital still cameras*. CRC press.

[143] Niebert, N., Baucke, S., El Khayat, I., Johnsson, M., Ohlman, B., Abramowicz, H., Wuenstel, K., Woesner, H., Quittek, J., and Correia, L. M. (2008). The way 4ward to the creation of a future internet. In *PIMRC*, pages 1–5.

[144] Ning, H. and Wang, Z. (2011). Future internet of things architecture: like mankind neural system or social organization framework? *IEEE Communications Letters*, 15(4):461–463.

[145] Nitti, M., Girau, R., Atzori, L., Iera, A., and Morabito, G. (2012). A subjective model for trustworthiness evaluation in the social internet of things. In *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, pages 18–23. IEEE.

[146] Palattella, M. R., Accettura, N., Grieco, L. A., Boggia, G., Dohler, M., and Engel, T. (2013a). On optimal scheduling in duty-cycled industrial iot applications using ieee802. 15.4 e tsch. *IEEE Sensors Journal*, 13(10):3655–3666.

[147] Palattella, M. R., Accettura, N., Vilajosana, X., Watteyne, T., Grieco, L. A., Boggia, G., and Dohler, M. (2013b). Standardized protocol stack for the internet of (important) things. *IEEE communications surveys & tutorials*, 15(3):1389–1406.

[148] Pandya, H. B. and Champaneria, T. A. (2015). Internet of things: Survey and case studies. In *Electrical, Electronics, Signals, Communication and Optimization (EESCO), 2015 International Conference on*, pages 1–6. IEEE.

[149] Peña-López, I. et al. (2005a). Itu internet report 2005: the internet of things.

[150] Peña-López, I. et al. (2005b). Itu internet report 2005: the internet of things.

[151] Perera, C., Zaslavsky, A., Christen, P., and Georgakopoulos, D. (2014). Context aware computing for the internet of things: A survey. *IEEE communications surveys & tutorials*, 16(1):414–454.

[152] Pintus, A., Carboni, D., and Piras, A. (2012). Paraimpu: a platform for a social web of things. In *Proceedings of the 21st International Conference on World Wide Web*, pages 401–404. ACM.

[153] Presser, M., Barnaghi, P. M., Eurich, M., and Villalonga, C. (2009). The sensei project: Integrating the physical world with the digital world of the network of the future. *IEEE Communications Magazine*, 47(4):1–4.

[154] Preuveneers, D. and Berbers, Y. (2008). Internet of things: A context-awareness perspective. *The Internet of Things: From RFID to the Next-Generation Pervasive Networked Systems*, pages 287–307.

[155] Prodan, R. and Ostermann, S. (2009). A survey and taxonomy of infrastructure as a service and web hosting cloud providers. In *Grid Computing, 2009 10th IEEE/ACM International Conference on*, pages 17–25. IEEE.

[156] Razzaque, M. A., Milojevic-Jevric, M., Palade, A., and Clarke, S. (2016). Middleware for internet of things: A survey. *IEEE Internet of Things Journal*, 3(1):70–95.

[157] Römer, K., Schoch, T., Mattern, F., and Dübendorfer, T. (2004). Smart identification frameworks for ubiquitous computing applications. *Wireless Networks*, 10(6):689–700.

[158] Romero, D., Hermosillo, G., Taherkordi, A., Nzekwa, R., Rouvoy, R., and Eliassen, F. (2010). Restful integration of heterogeneous devices in pervasive environments. In *IFIP International Conference on Distributed Applications and Interoperable Systems*, pages 1–14. Springer.

[159] Roselli, L., Mariotti, C., Mezzanotte, P., Alimenti, F., Orecchini, G., Virili, M., and Carvalho, N. (2015). Review of the present technologies concurrently contributing to the implementation of the internet of things (iot) paradigm: Rfid, green electronics, wpt and energy harvesting. In *Wireless Sensors and Sensor Networks (WiSNet), 2015 IEEE Topical Conference on*, pages 1–3. IEEE.

[160] Rouvoy, R., Barone, P., Ding, Y., Eliassen, F., Hallsteinsen, S., Lorenzo, J., Mamelli, A., and Scholz, U. (2009). Music: Middleware support for self-adaptation in ubiquitous and service-oriented environments. In *Software engineering for self-adaptive systems*, pages 164–182. Springer.

[161] S. Cheshire, M. K. (2011). Dns-based service discovery.

[162] Saha, D., MukheJee, A., and Biswas, C. (2006). Present scenarios and future challenges in pervasive middleware. *IEEE Pervasive Computing*.

[163] Saint-Andre, P. (2011). Extensible messaging and presence protocol (xmpp): Core. Technical report.

[164] Sammarco, C. and Iera, A. (2012). Improving service management in the internet of things. *Sensors*, 12(9):11888–11909.

[165] Schneider, S. et al. (2013). Understanding the protocols behind the internet of things. *Electronic Design*, 9(10).

[166] Semmelhack, P. (2013). *Social machines: how to develop connected products that change customers' lives.* John Wiley & Sons.

[167] Shamszaman, Z. U. and Ali, M. I. (2017). Towards a smart society through semantic virtual-object enabled real-time management framework in the social internet of things. *IEEE Internet of Things Journal*.

[168] Shamszaman, Z. U. and Ali, M. I. (2018). Enabling cognitive contributory societies using siot: Qos aware real-time virtual object management. *Journal of Parallel and Distributed Computing*.

[169] Shamszaman, Z. U., Ali, M. I., and Mileo, A. (2015). On the need for adaptivity in rdf stream processing.

[170] Shamszaman, Z. U., Ara, S. S., Chong, I., and Jeong, Y. K. (2014). Web-of-objects (woo)-based context aware emergency fire management systems for the internet of things. *Sensors*, 14(2):2944–2966.

[171] Shi, Y. and Chen, X. (2011). A survey on qos-aware web service composition. In *Multimedia Information Networking and Security (MINES), 2011 Third International Conference on*, pages 283–287. IEEE.

[172] Spenza, D., Petrioli, C., and Cammarano, A. (2012). Pro-energy: A novel energy prediction model for solar and wind energy-harvesting wireless sensor networks. In *2012 IEEE 9th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS 2012)*, pages 75–83. IEEE.

[173] Spiess, P., Karnouskos, S., Guinard, D., Savio, D., Baecker, O., De Souza, L. M. S., and Trifa, V. (2009). Soa-based integration of the internet of things in enterprise services. In *Web Services, 2009. ICWS 2009. IEEE International Conference on*, pages 968–975. IEEE.

[174] Stojmenovic, I. and Wen, S. (2014). The fog computing paradigm: Scenarios and security issues. In *Computer Science and Information Systems (FedCSIS), 2014 Federated Conference on*, pages 1–8. IEEE.

[175] Sundmaeker, H., Guillemin, P., Friess, P., and Woelfflé, S. (2010). Vision and challenges for realising the internet of things. *Cluster of European Research Projects on the Internet of Things, European Commision*, 3(3):34–36.

[176] Suo, H., Wan, J., Zou, C., and Liu, J. (2012). Security in the internet of things: a review. In *Computer Science and Electronics Engineering (ICCSEE), 2012 international conference on*, volume 3, pages 648–651. IEEE.

[177] Swan, M. (2012). Sensor mania! the internet of things, wearable computing, objective metrics, and the quantified self 2.0. *Journal of Sensor and Actuator Networks*, 1(3):217–253.

[178] Tan, J. and Koo, S. G. (2014). A survey of technologies in internet of things. In *2014 IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS)*, pages 269–274. IEEE.

[179] Tan, L. and Wang, N. (2010). Future internet: The internet of things. In *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, volume 5, pages V5–376. IEEE.

[180] Terziyan, V., Kaykova, O., and Zhovtobryukh, D. (2010). Ubiroad: Semantic middleware for context-aware smart road environments. In *Internet and web applications and services (iciw), 2010 fifth international conference on*, pages 295–302. IEEE.

[181] Tsai, W. T., L. Y. H. C. Z. C. Y. X. B. (2006). Rtsoa: Real-time service-oriented architecture. *Second IEEE International Symposium on Service-Oriented System Engineering (SOSE'06),*, pages 49–56.

[182] Vazquez, J. I. and Lopez-De-Ipina, D. (2008). Social devices: autonomous artifacts that communicate on the internet. In *The Internet of Things*, pages 308–324. Springer.

[183] Vermesan, O. and Friess, P. (2013). *Internet of things: converging technologies for smart environments and integrated ecosystems*. River Publishers.

[184] Vlacheas, P., Giaffreda, R., Stavroulaki, V., Kelaidonis, D., Foteinos, V., Poulios, G., Demestichas, P., Somov, A., Biswas, A. R., and Moessner, K. (2013). Enabling smart cities through a cognitive management framework for the internet of things. *IEEE communications magazine*, 51(6):102–111.

[185] Vogt, H. (2002). Efficient object identification with passive rfid tags. In *International Conference on Pervasive Computing*, pages 98–113. Springer.

[186] Wang, H., Zhou, X., Zhou, X., Liu, W., Li, W., and Bouguettaya, A. (2010). Adaptive service composition based on reinforcement learning. In *International Conference on Service-Oriented Computing*, pages 92–107. Springer.

[187] Want, R., Fishkin, K. P., Gujar, A., and Harrison, B. L. (1999). Bridging physical and virtual worlds with electronic tags. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*, pages 370–377. ACM.

[188] Wei, W., Yang, A. T., and Shi, W. (2016). Security in internet of things: Opportunities and challenges. In *Identification, Information and Knowledge in the Internet of Things (IIKI), 2016 International Conference on*, pages 512–518. IEEE.

[189] Wellman, B. et al. (1997). Structural analysis: From method and metaphor to theory and substance. *Contemporary Studies in Sociology*, 15:19–61.

[190] Whitmore, A., Agarwal, A., and Da Xu, L. (2015). The internet of things—a survey of topics and trends. *Information Systems Frontiers*, 17(2):261–274.

[191] Winter, Tim, e. a. (2002). Efficient energy sensitive routing protocols in mobile ad-hoc networks. pages 475–480.

[192] Wu, F.-J., Kao, Y.-F., and Tseng, Y.-C. (2011). From wireless sensor networks towards cyber physical systems. *Pervasive and Mobile computing*, 7(4):397–413.

[193] Wu, J. and Zhao, W. (2017). Design and realization of winternet: From net of things to internet of things. *ACM Transactions on Cyber-Physical Systems*, 1(1):2.

[194] Wu, M., Lu, T.-J., Ling, F.-Y., Sun, J., and Du, H.-Y. (2010). Research on the architecture of internet of things. In *Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on*, volume 5, pages V5–484. IEEE.

[195] Xiao, Y., Rayi, V. K., Sun, B., Du, X., Hu, F., and Galloway, M. (2007). A survey of key management schemes in wireless sensor networks. *Computer communications*, 30(11-12):2314–2341.

[196] Yan, L., Zhang, Y., Yang, L. T., and Ning, H. (2008). *The Internet of things: from RFID to the next-generation pervasive networked systems.* CRC Press.

[197] Yang, L., Li, W., Ghandehari, M., and Fortino, G. (2018). People-centric cognitive internet of things for the quantitative analysis of environmental exposure. *IEEE Internet of Things Journal*, 5(4):2353–2366.

[198] Yang, X., Lin, J., Yu, W., Moulema, P.-M., Fu, X., and Zhao, W. (2015). A novel en-route filtering scheme against false data injection attacks in cyber-physical networked systems. *IEEE Transactions on Computers*, 64(1):4–18.

[199] Youssef, T. A., Elsayed, A. T., and Mohammed, O. A. (2016). Data distribution service-based interoperability framework for smart grid testbed infrastructure. *Energies*, 9(3):150.

[200] Yu, W., J. L. (2012). Rpl: Ipv6 routing protocol for low-power and lossy networks.

[201] Yu, W., Le, T. N., Xuan, D., and Zhao, W. (2004). Query aggregation for providing efficient data services in sensor networks. In *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pages 31–40. IEEE.

[202] Yu, W., Xuan, D., Graham, B., Santhanam, S., Bettati, R., and Zhao, W. (2002). An integrated middleware-based solution for supporting secured dynamic-coalition applications in heterogeneous environments. *Submitted to IEEE Tran. On Systems, Man, and Cybernetics.*

[203] Zanella, A., Bui, N., Castellani, A., Vangelista, L., and Zorzi, M. (2014). Internet of things for smart cities. *IEEE Internet of Things journal*, 1(1):22–32.

[204] Zeng, L., Benatallah, B., Dumas, M., Kalagnanam, J., and Sheng, Q. Z. (2003). Quality driven web services composition. In *Proceedings of the 12th international conference on World Wide Web*, pages 411–421. ACM.

[205] Zeng, L., Benatallah, B., Ngu, A. H., Dumas, M., Kalagnanam, J., and Chang, H. (2004a). Qos-aware middleware for web services composition. *IEEE Transactions on software engineering*, 30(5):311–327.

[206] Zeng, L., Benatallah, B., Ngu, A. H., Dumas, M., Kalagnanam, J., and Chang, H. (2004b). Qos-aware middleware for web services composition. *IEEE Transactions on software engineering*, 30(5):311–327.

[207] Zhang, C., Cheng, C., and Ji, Y. (2012a). Architecture design for social web of things. In *Proceedings of the 1st International Workshop on Context Discovery and Data Mining*, page 3. ACM.

[208] Zhang, Y., Pham, M.-D., Corcho, Ó., and Calbimonte, J.-P. (2012b). Srbench: A streaming rdf/sparql benchmark. In *ISWC (1)*, pages 641–657.

[209] Zhang Chengwen, Su Sen, C. J. (2006a). A novel genetic algorithm for qos-aware web services selection. *Data Engineering Issues in E-Commerce and Services.*

[210] Zhang Chengwen, Su Sen, C. J. (2006b). Qos-aware web service selection based on genetic algorithm. *Chinese Journal of Computers.*

[211] Zhao, K. and Ge, L. (2013). A survey on the internet of things security. In *Computational Intelligence and Security (CIS), 2013 9th International Conference on*, pages 663–667. IEEE.

[212] Zhenhua, M., Hongming, C., and Lihong, J. (2009). Service selection problem with multiple qos constraints based on genetic algorithm. *Computer Applications and Software*, 26(3):4–6.

[213] Zhou, H. (2012a). *The internet of things in the cloud: A middleware perspective.* CRC press.

[214] Zhou, H. (2012b). *The internet of things in the cloud: A middleware perspective.* CRC press.

[215] Zorzi, M., Gluhak, A., Lange, S., and Bassi, A. (2010). From today's intranet of things to a future internet of things: a wireless-and mobility-related view. *IEEE Wireless communications*, 17(6).