Edward Curry

# Real-time Linked Dataspaces

## Enabling Data Ecosystems for Intelligent Systems

Springer Open

# Real-time Linked Dataspaces

Edward Curry

# Real-time Linked Dataspaces

## Enabling Data Ecosystems for Intelligent Systems

Springer Open

Edward Curry
National University of Ireland Galway
Galway, Ireland

ISBN 978-3-030-29664-3 ISBN 978-3-030-29665-0 (eBook)
https://doi.org/10.1007/978-3-030-29665-0

This book is an open access publication.

© The Editor(s) (if applicable) and The Author(s) 2020
**Open Access** This book is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.
The images or other third party material in this book are included in the book's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the book's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.
The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.
The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

*To Meg, Liam, and Roisin*

*To Mum and Dad*

# Foreword

In Lewis Carroll's *Through the Looking-Glass*, the Red Queen explained to Alice the nature of Looking-Glass Land *"Now, here, you see, it takes all the running you can do, to keep in the same place."* Van Valen coined the "Red Queen" hypothesis where populations have to "run" an evolutionary race in order to stay in the same place, or else go extinct. Within our Digital Universe, we have a Digital Red Queen, with a race between our ability to create masses of data and our ability to manage it effectively and efficiently.

Over the last quarter of a century, we have both run this race with our work in the corporate worlds of Google and Verizon, to the ivory towers of MIT and the University of Washington. We were both extremely interested in the problems of data integration at scale within ecosystems and have proposed approaches for doing so. Naturally, we felt more than a little curious to see what Ed and his team had produced.

In the decade since our original works on ecosystems and dataspaces, we have seen new data management needs arise from the mass migration of applications from batch processing paradigms to real-time processing. This sets the scene for the book as it introduces "Real-time" dataspaces to enable data flows within ecosystems of intelligent systems. Within these covers, you will find a technical vision, new techniques, and deep insight for both theory and practice of dataspaces for real-time data. The book brings us on a journey from the lab to the field by developing new pioneering best-effort techniques for real-time data management and validates their use within an excellent choice of an application domain, not just resource management but specifically sustainability. This body of work illustrates how the dataspace paradigm has evolved, and the transformative potential of leveraging data ecosystems to drive value within intelligent systems. The work goes beyond a purely technical perspective and exposes the critical social and organizational aspects of managing data ecosystems for the collective benefit of the participants.

We are delighted to write this foreword to a book that will influence the thinking on the design of data infrastructures as a key enabler of data ecosystems, intelligent systems, and smart environments. It sets out a clear path for the design of data platforms based on dataspaces with support for best-effort real-time data processing techniques. This impressive body of work illustrates the power that data-driven systems have to improve the sustainability of our planet's complex ecosystems, both Natural and Digital.

MIT, Cambridge, MA, USA Michael L. Brodie
Facebook AI, Menlo Park, CA, USA Alon Halevy
August 2019

# Preface

Around 2012 I started to investigate the potential of data-driven intelligent systems for sustainability. I was (and still am) very motivated by the potential of the Internet of Things (IoT), data analytics, and artificial intelligence to create intelligent systems that can contribute to a sustainable society. As a computer scientist with a background in distributed systems and data management, I felt I could make a modest contribution to the design and construction of these intelligent systems. There was significant potential for data-driven and artificial intelligence techniques to power intelligent systems for sustainability. However, for these approaches to be viable, they would need to be cost-effective and deployable. Working with my industrial collaborations, it was clear that a critical barrier to the adoption of intelligent systems was, and still is, the high upfront costs associated with data sharing and integration. For decades we have seen the consequences of data silos within Enterprises with estimates of 50–80% of the costs of data projects going to data integration and preparation activities. This limits large-scale data management projects to large organisations that have the necessary expertise and resources. This needs to change if we want a broad effort for sustainability that enables smaller stakeholders to engage and leverage the value available in data.

Datafication driven by IoT-based digital infrastructure is leading to an ecosystem of data which can be exploited to transform our world. Typically, IoT data has the most value when it can be processed on-the-fly and with low-latency. However, the current wave of datafication is leading to increasing "data silos." In 2012, the IoT was predicted to have 25 billion connected devices by 2020; current estimates are now for over 75 billion connected devices by 2025. What was evident in 2012, and is even more apparent today, is that we need a fundamental transformation in how we manage the data ecosystem surrounding intelligent systems in smart environments. Traditional approaches to data management will not be sufficient. We need a paradigm shift as significant as the move to relational data management in the 1970s to provide an alternative to the current top-down, centralised models of data management.

In the first two decades of the twenty-first century, a recognition emerged among researchers and practitioners that a new class of information management and processing systems was needed to support diverse distributed real-time applications. Michael Brodie has been a Prophet of the data integration challenges within ecosystems where thousands of semantically heterogeneous databases need to be managed and integrated collectively. These information ecosystems necessitate a transformation in how data is managed and shared among intelligent systems. Halevy, Franklin, and Maier recognised that in large-scale integration scenarios, involving thousands of data sources (such as ecosystems), it is difficult and expensive to obtain an upfront unifying schema across all sources. They introduced the paradigm of Dataspaces that shifts the emphasis to providing support for the co-existence of heterogeneous data that does not require a significant upfront investment into a unifying schema. The concepts of ecosystems and dataspaces were absorbing, and I was excited by the potential of “best-effort” approaches. I felt there might be a possible connection to the Pareto Principle.

The Pareto Principle (or the 80/20 rule) has wide application in many areas from economics and market analysis to business strategy, where it has been observed that 20% of the effort delivers 80% of the results. Within computer science, this principle has been observed within many problems from fixing bugs to writing code. The principle can help us to prioritise actions, for example, focus on the 20% of software bugs that cause 80% of the system crashes. The power of the principle has always fascinated me, and the dataspaces paradigm can unlock its power within the data realm. The pay-as-you-go model allows participants in the dataspace to focus on high-value data and tackle the “long tail” of data on an as-needed basis. This was the genesis of this work.

This book explores the dataspace paradigm as an alternative best-effort approach to data management with data ecosystems. It establishes the theoretical foundations and principles of Real-time Linked Dataspaces as a data platform for intelligent systems, and introduces a set of specialised best-effort techniques and models to enable loose administrative proximity and semantic integration for managing and processing events and streams.

Readers of this book will gain a detailed understanding of how the dataspace paradigm is used to enable data ecosystems for intelligent systems within smart environments. The reader is brought from establishing the fundamental theory and the creation of new techniques needed for support services, to the experience gained from delivering real-world intelligent systems for smart cities, buildings, energy, water, and mobility.

The book is of interest to three key audiences. First are researchers and graduate students in the fields of data management, big data, IoT, and intelligent systems with interest in state-of-the-art techniques for approximate and best-effort approaches to incremental data management. Second, the book provides useful insights to practitioners that need to create advanced data management platforms for intelligent systems, smart environments, and data ecosystems. Practitioners will learn about designing incremental data management architectures and techniques that are grounded in theory and informed by the experience of rigorous deployments within real-world settings. Third, researchers and practitioners involved in interdisciplinary and transdisciplinary “Smart” projects will gain insights on the design and operation of data-intensive socio-technical intelligent systems.

The book is structured as follows: *Part I: Fundamentals and Concepts* details the motivation and core concepts of Real-time Linked Dataspaces. This part establishes the need for an evolution of data management techniques to meet the challenges of enabling a data ecosystem for intelligent systems within smart environments. It details the fundamental concepts of Dataspaces and the need for specialisation for processing dynamic real-time data. *Part II: Data Support Services* explores the design and evaluation of critical services within the Real-time Linked Dataspace, including catalog, entity management, query and search, data service discovery, and human-in-the-loop. *Part III: Stream and Event Processing Services* details the design and evaluation of the specialised techniques created for real-time support services including complex event processing, event service composition, stream dissemination, stream matching, and approximate semantic matching. *Part IV: Intelligent Systems and Applications* explores the use of Real-time Linked Dataspaces within real-world smart environments by demonstrating its role in enabling intelligent water and energy management systems through the development of IoT-enabled digital twins, enhanced user experience, and autonomic source selection for advanced predictive analytics. Finally, *Part V: Future Directions* details research challenges for dataspaces, data ecosystems, and intelligent systems.

Forward-thinking societies will see the provision of digital infrastructure as a shared societal service in the same way as water, sanitation, and healthcare. With few exceptions, our current large-scale data infrastructures are beyond the reach of small organisations who cannot deal with the complexity of data management and the high costs associated with data infrastructure. It is clear we desperately need new approaches to support the complex data ecosystems our “smart” society is creating. This vision demands a fundamental shift in how to design large-scale data ecosystem infrastructure to unlock the power of a Pareto effect for data. I believe this book is a step in that direction.

Galway, Ireland Edward Curry
October 2019

# Acknowledgements

I want to thank all my fantastic students André, Feng, Souleiman, Umair, Wassim, Eanna, Ninad, and Willem who have shared this journey with me as we created many of the ideas and concepts presented in this book. It has been an honour for me to be their guide on the start of their scientific journey, and I am richer for the experience. My collaborators have been a critical part of this work. I want to thank Yongrui "Louie" Qin, Quan Z. "Michael" Sheng, and the project teams from Waternomics and SENSE. In particular, I would like to thank Sean and Adeboyega for their encouragement and helpful discussion over the years. I want to thank Stefan, Manfred, Dietrich, and Mathieu for creating and nurturing a fantastic environment for ideas to germinate and blossom into life, and for always encouraging us to think big, be brave, and deliver impact.

I gratefully acknowledge the funders of this work; it was supported in part by Science Foundation Ireland (SFI) under Grant Number SFI/12/RC/2289, the European Commission's Seventh Framework Programme from ICT grant agreement WATERNOMICS no. 619660, Enterprise Ireland, the National Development Plan, and European Union under Grant Number IP/2012/0188.

I want to thank Niki, Asra, Piyush, Tarek, Atiya, Amin, and Edobor for their support in the preparation of the final manuscript. Thanks also go to Ralf Gerstner and all at Springer for their professionalism and assistance.

I want to thank my parents Edward and Mary, my sisters Tara and Alison, brother Joe, and my parents-in-law Hilda and Roger. Finally, and most importantly I would like to thank my wife, Meg, for her support, encouragement, and unwavering love; my son Liam for all the "Great Ideas" he gave his daddy, and to Roisin who bloomed into a beautiful little rose as I wrote this book. You make everything worthwhile.

Galway, Ireland Edward Curry

# Contents

Part I Fundamentals and Concepts

**Part II Data Support Services**

**Part III Stream and Event Processing Services**

**Part IV Intelligent Systems and Applications**

**Part V Future Directions**

# Part I
# Fundamentals and Concepts

The first part of this book details the motivation and core concepts of Real-time Linked Dataspaces. This part establishes the need for an evolution of data management techniques to meet the challenges of data ecosystems for intelligent systems. It details the fundamental concepts of dataspaces and the need for a data platform for intelligent systems within Internet of Things-based smart environments.

# Chapter 1
# Real-time Linked Dataspaces: A Data Platform for Intelligent Systems Within Internet of Things-Based Smart Environments

**Keywords** Data platform · Intelligent systems · Internet of things · Data ecosystem · Smart environment · Dataspace

## 1.1 Introduction

Around 18,000 BCE, Paleolithic tribespeople marked notches into sticks, or bones, to keep track of trading activity or supplies. The tribespeople would compare the notches on their prehistoric data storage devices (their tally sticks) to make basic calculations that would allow them to make predictions such as how long their food supplies would last. From these early examples, we can trace a gradual evolution of the ability of humans to store, analyse, and share information.

In the first decades of the twenty-first century, datafication is driving the transformation of our everyday world, from the digitisation of traditional infrastructure (smart energy, water, and mobility) to the revolution of industrial sectors (cyber-physical systems, autonomous vehicles, and Industry 4.0), and changes to how our society works (smart government and cities). The contemporary wave of datafication is creating smart environments that are powered by digital technologies such as the Internet of things, big data, and artificial intelligence. Within these smart environments, intelligent systems are creating data ecosystems with unprecedented levels of real-time data about our world [1]. A recognition has emerged among researchers and practitioners that a new class of information management and processing systems is needed to support diverse distributed real-time data-intensive intelligent systems. These applications necessitate a transformation in how data is managed and shared among systems [2] and in how data can be processed on-the-fly and with low-latency [3]. Both of these requirements are critical if we are to extract the maximum value from the current wave of datafication, and both topics have a rich body of ongoing work. However, there is a paucity of research on approaches that tackle both of these requirements together for the large-scale sharing of real-time data.

Real-time Linked Dataspaces (RLD) address this need by combining pay-as-you-go data management with techniques for flexible data integration and real-time

© The Author(s) 2020

E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_1

processing and query. This book establishes the foundations and principles of Real-time Linked Dataspaces [4] as a data platform for intelligent systems within smart environments. It investigates the “best-effort” approximate techniques needed to process real-time data within the dataspace paradigm. The book details state-of-the-art techniques from artificial intelligence, knowledge graphs, Internet of things, and advanced stream and event processing to complement the dataspace approach to effectively and efficiently manage and extract value from data within Internet of things-based smart environments.

The remainder of this chapter is structured as follows: Sect. 1.2 begins by establishing the foundations of Real-time Linked Dataspaces with an overview of intelligent systems, smart environments, Internet of things, data ecosystems, and the need for a data platform. Section 1.3 introduces the notion of a Real-time Linked Dataspace and its role as a data platform for intelligent systems within smart environments. An overview of the structure of the book is provided in Sect. 1.4, with a summary in Sect. 1.5.

## 1.2 Foundations

As illustrated in Fig. 1.1, Real-time Linked Dataspaces lie at the intersection of the fields of Data Management (data ecosystems, pay-as-you-go, knowledge graphs), Distributed Systems (Internet of things, event and stream processing), Artificial Intelligence (intelligent systems), and Ubiquitous Computing (smart environments). These fields of computer science need to be brought together to enable breakthroughs not possible when the fields work in isolation. In this section, we examine the recent developments in these fields with the rise in importance of data-intensive techniques and the need to support data sharing between the ecosystems of interconnected intelligent systems within Internet of Things (IoT)-based smart environments.

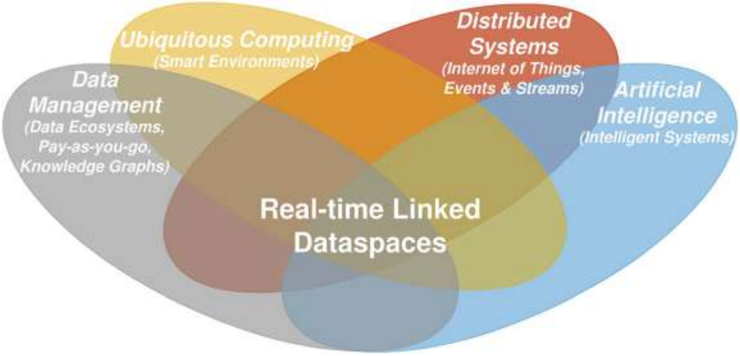


**Fig. 1.1** Foundations of Real-time Linked Dataspaces

### 1.2.1 Intelligent Systems

Originating from the field of Artificial Intelligence (AI), intelligent systems are revolutionising many industries and society, including transportation and logistics, security, manufacturing, energy, healthcare, and agriculture, by providing the “built-in” intelligence to improve efficiency, quality, and flexibility. An intelligent system can gather, represent, reason, and interpret data. In doing so, it can learn, extract patterns and meaning, derive new information, learn from experience, and identify strategies and behaviours to act intelligently. Contemporary intelligent systems are usually Internet-connected with an ability to communicate and collaborate with other systems. Several definitions for intelligent systems exist with some of these captured in Table 1.1.

Intelligent systems are complex and can be created using a wide range of techniques from AI, machine learning (supervised, unsupervised, and reinforcement learning), deep learning, computer vision, natural language processing, to complex event processing, and knowledge graphs. The inspiration for the design of intelligent systems is often drawn from ideas and concepts from nature’s problem-solving approaches across a range of fields, including biology, cognitive science, and neuroscience which results in many interdisciplinary relationships. The design and construction of intelligent systems is a vibrant area of active research, which is the subject of many excellent books. The focus of this book is to support the sharing of data between intelligent systems within an IoT-enabled smart environment.

### 1.2.2 Smart Environments

Smart environments have evolved from the fields of ubiquitous and pervasive computing that promote the idea of an information communication technology-

**Table 1.1** Definitions of an Intelligent System

| Definition | References |
|---|---|
| “An intelligent system is one that uses artificial intelligence (AI) techniques to offer important services (e.g., as a component of a larger system) to allow integrated systems to perceive, reason, learn, and act intelligently in the real world.” | ACM Transactions on Intelligent Systems and Technology |
| “Intelligent systems perform search and optimization along with learning capabilities.” | [5] |
| “Intelligent systems connect users to artificial intelligence (machine learning) to achieve meaningful objectives. An intelligent system is one in which the intelligence evolves and improves over time, particularly when the intelligence improves by watching how users interact with the system.” | [6] |
| “A system, which is based on approach(es), method(s) or technique (s) of the artificial intelligence field to perform more accurate and effective operations for solving the related problems.” | [7] |

**Table 1.2** Definitions of a Smart Environment

| Definition | References |
|---|---|
| "On a physical world richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives and connected through a continuous network." | [8] |
| "An ecosystem of interacting objects, e.g. sensors, devices, appliances and embedded systems in general, that have the capability to self-organize, to provide services and manipulate/publish complex data." | [9] |
| "A physical world interwoven with invisible sensors, actuators, displays, and computational elements. These computing elements are generally embedded seamlessly in everyday objects and networked to each other and beyond (the internet, usually)." | [10] |
| "One that is able to acquire and apply knowledge about the environment and its inhabitants in order to improve their experience in that environment." | [11, 12] |

enabled physical world. Mark Weiser defined a smart environment as "a physical world that is richly and invisibly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the everyday objects of our lives, and connected through a continuous network" [8]. Several definitions for smart environments are detailed in Table 1.2. These definitions illustrate the complexity of the challenge associated with delivering a smart environment. Their realisation needs contributions from several research fields to be brought together to deliver on this vision, including distributed computing, mobile computing, location computing, context-aware computing, wireless sensor networks, human–computer interaction, ambient intelligence, and artificial intelligence.

In the past decade, smart environments have started to move from a research vision to concrete manifestations in real-world deployments. As smart environments are realised, they encounter a number of practical challenges, including the interoperability of diverse technology (e.g. legacy systems) [13], meeting the needs of diverse stakeholders with very broad goals and expectations, and working within the limited budgets available to invest in infrastructure. The understanding of these challenges in more detail requires an understanding of how the IoT is enabling real-time data processing within smart environments.

### *1.2.3 Internet of Things*

A key driver in the development of smart environments is the convergence of technologies such as the IoT and big data, which is driving the digitisation of physical infrastructures with sensors, networks, and social capabilities [14]. The vision of the IoT is a complicated proposition that requires end-to-end distributed systems from the development of new electronic devices and embedded systems, new forms of data processing to deal with the volume, variety, and velocity of data generated, to enhanced user experiences leveraging cognitive and behavioural models with new data visualisation and interaction paradigms.

As the IoT enables the deployment of lower-cost sensors, we see more broad adoption of IoT devices/sensors and gain more visibility (and data) into smart environments. This results in high volume and high-velocity event streams from smart environments that need to be processed. IoT-based smart environments are also generating different types of data with an increase in the number of multimedia devices deployed, such as vehicle and traffic cameras. The IoT is driving the deployment of intelligent systems and creating new opportunity in smart environments:

- **Digital Twins**: A digital replica of physical assets (car), processes (value-chain), systems, or physical environments (building). The digital representation (i.e. simulation modelling or data-driven model) provided by the digital twin can be analysed to optimise the operation of the "physical twin".
- **Physical-Cyber-Social (PCS)**: A computing paradigm that supports a richer human experience with a holistic data-rich view of the smart environment that integrates, correlates, interprets, and provides contextually relevant abstractions to humans [14].
- **Mass Personalisation**: More human-centric thinking in the design of systems where users have growing expectations for highly personalised digital services for the "Market of One".
- **Data Network Effects**: As more systems/users join and contribute data to the smart environment, a "network effect" can take place, resulting in the overall data available becoming more valuable.

Within this context, we are interested in how data created within a smart environment can be leveraged by intelligent systems, and how data can be easily shared within the ecosystem of systems (new and old) and stakeholders.

### *1.2.4 Data Ecosystems*

A Data Ecosystem is a socio-technical system enabling value to be extracted from data value chains supported by interacting organisations and individuals [15]. Within an ecosystem, data value chains are oriented to business and societal purposes. The ecosystem can create the conditions for a marketplace competition among participants or enable collaboration among diverse, interconnected participants that depend on each other for their mutual benefit.

The digital transformation is creating a data ecosystem with data on every aspect of our world, spread across a range of intelligent systems. As illustrated in Fig. 1.2, a smart environment enabled with IoT data, and contextual data sources, results in a data-rich ecosystem of structured and unstructured data (e.g. images, video, audio, and text) that can be exploited by data-driven intelligent systems.

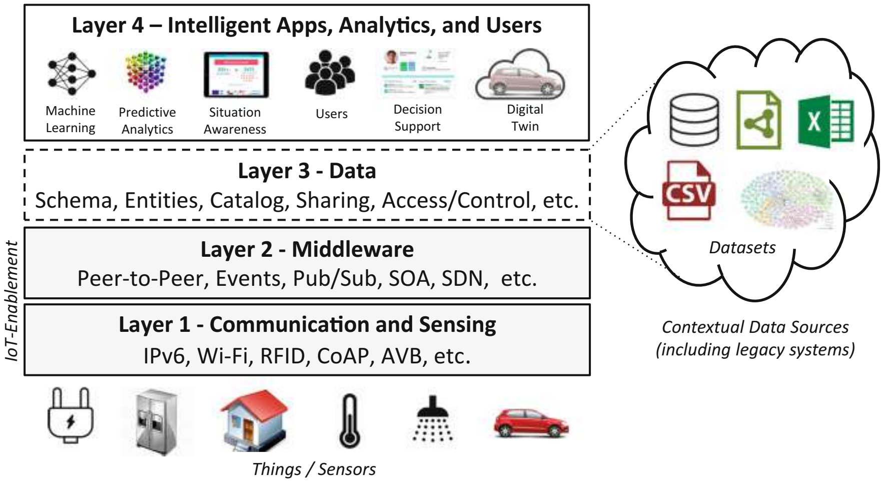

Layer 4 – Intelligent Apps, Analytics, and Users
Machine Learning
Predictive Analytics
Situation Awareness
Users
Decision Support
Digital Twin
Layer 3 - Data
Schema, Entities, Catalog, Sharing, Access/Control, etc.
Layer 2 - Middleware
Peer-to-Peer, Events, Pub/Sub, SOA, SDN, etc.
Layer 1 - Communication and Sensing
IPv6, Wi-Fi, RFID, CoAP, AVB, etc.
IoT-Enablement
Things / Sensors
CSV
Datasets
Contextual Data Sources
(including legacy systems)

**Fig. 1.2** Four-layered framework to enable data ecosystems for intelligent systems within IoT-based smart environments [4]

There is a need to bring together data from the multiple intelligent systems that exist within the data ecosystem surrounding a smart environment. For example, smart cities are showing how different systems within the city (e.g. energy and transport) can collaborate to maximise the potential to optimise overall city operations. At the level of an individual, digital services can deliver a personalised and seamless user experience by bringing together relevant user data from multiple systems [16]. This requires a System of Systems (SoS) approach to connect systems that cross organisational boundaries, come from various domains (e.g. finance, manufacturing, facilities, IT, water, traffic, and waste) and operate at different levels (e.g. region, district, neighbourhood, building, business function, individual).

Data ecosystems present new challenges to the design of intelligent systems and SoS that require a rethink in how we should deal with the needs of large-scale, data-rich smart environments. How can we support data sharing between intelligent systems in a data ecosystem? What are the technical and non-technical barriers to data sharing within the ecosystem? How can intelligent systems leverage their data ecosystem to be “smarter”? Solving these problems is critical if we are to maximise the potential of data-intensive intelligent systems [1].

### *1.2.5 Enabling Data Ecosystem for Intelligent Systems*

Understanding the data management challenges in more detail requires an appreciation of how the IoT is enabling smart environments. The range of IoT challenges can be studied based on the three-layered framework by Atzori et al. [17] *Layer 1—Communication and Sensing; Layer 2—Middleware; and Layer 3—Users, Applications, & Analytics*. At the data level, intelligent systems can benefit from leveraging data from multiple systems within the smart environment. However, many of the data management and sharing activities are currently performed at the application layer within IoT deployments. We elaborate a four-layered framework to enable data ecosystems for intelligent systems within IoT-based smart environments that builds on the work by Atzori et al. [17]. As illustrated in Fig. 1.2, we introduce a fourth layer between the Middleware and Application layers to support data management and sharing activities. The four-layered framework for enabling data ecosystems for intelligent systems consists of:

- **Layer 1—Communication and Sensing**: An essential requirement is an infrastructure of communication and sensing that maps the world of physical things into the world of computationally processable data.
- **Layer 2—Middleware**: Middleware abstracts the application developers from the underlying technologies. Data distribution, processing, and access to legacy information systems take place at this layer.
- **Layer 3—Data**: There is a need to enable data management and sharing activities, including managing schema and entities, accessibility, access control, data quality, and licensing take place at this layer.
- **Layer 4—Intelligent Applications, Analytics, and Users**: Users expect IoT-based analytics and applications that present the data gathered and analysed in an intuitive and user-friendly manner using new visualisations and user experiences to ensure cognitive-friendly smart environments.

Our key addition is *Layer 3—Data,* which requires the development of data infrastructure to support the sharing and management of data among systems in the ecosystem. Platform approaches have proved successful in many areas of technology, and the idea of large-scale "data" platforms are touted as a possible next step. A data platform focuses on secure and trusted data sharing amongst a group of participants (e.g. industrial consortiums sharing private or commercially sensitive data) within a clear legal framework. Within a smart environment, a data platform would support continuous, coordinated data flows, seamlessly moving data among intelligent systems.

## 1.3 Real-time Linked Dataspaces

In this book, we advocate the use of the dataspace paradigm to support the sharing of data between intelligent systems within IoT-enabled smart environments. The dataspace approach recognises that in large-scale integration scenarios, involving thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources [2]. Dataspaces are not a data integration approach [19]; they shift the emphasis to providing support for the co-existence of heterogeneous data that does not require a significant upfront investment into a unifying schema. Data is integrated on an "as-needed" basis with the labour-intensive aspects of data integration postponed until they are required. Dataspaces reduce the initial effort required to set up data integration by relying on automatic matching and mapping generation techniques. This results in a loosely integrated set of data sources. When tighter semantic integration is required, it can be achieved in an incremental "pay-as-you-go" fashion by detailed mappings among the required data sources.

The Real-time Linked Dataspace (RLD) is a platform for data management for intelligent systems within smart environments that combines the pay-as-you-go paradigm of dataspaces, linked data, and knowledge graphs with entity-centric real-time query capabilities [4]. In order to enable the dataspace principles to support real-time data processing, we created a specialised dataspace support service for loose administrative proximity and semantic integration for event and stream systems. This requirement forms the foundation of the techniques and models used to process events and streams within RLD.

The RLD contains all the relevant information within a data ecosystem including things, sensors, and data sources and has the responsibility for managing the relationships among these participants. The RLD goes beyond a traditional dataspace approach by supporting the management of entities within the data ecosystem as first-class citizens along with data sources, and it extends the dataspace support platform with real-time processing and querying capabilities. Figure 1.3 illustrates the architecture of the RLD with the following central concepts:

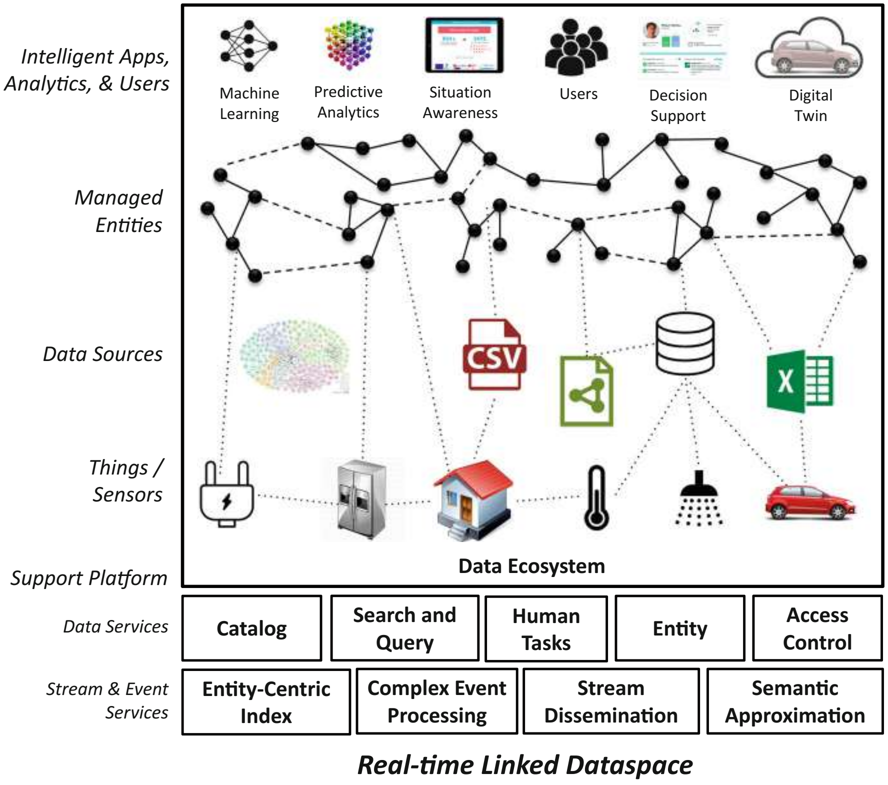


**Fig. 1.3** Real-time Linked Dataspace architecture [4]

- **Support Platform**: Responsible for providing the functionalities and services essential for managing the dataspace. Support services are grouped into data services and stream and event services.
- **Things/Sensors**: Produce real-time data streams that need to be processed and managed. Things in a smart environment range from connected personal devices and sensors to connected cars and manufacturing equipment.
- **Data Sources**: Data can be available in a wide variety of formats and accessible through different system interfaces. Some examples of data sources include building management systems, energy and water management systems, personal information systems, enterprise databases, weather forecasts, and (linked) open data.
- **Managed Entities**: Actively managed entities within the data ecosystem, including their relationship to participating things, data sources, and other entities in the RLD.
- **Intelligent Applications, Analytics, and Users**: Interact with the RLD and leverage its data and services to provide data analytics, decision support tools, user interfaces, and data visualisations. Applications/users can query the RLD in an entity-centric manner, while users can be enlisted in the curation of the data and entities via the Human Task service.

The RLD has been used as a data platform to support the development of intelligent applications within a range of IoT-based smart environments including smart home, school, office building, university, and airport [16]. Within these environments, a data platform needs to support a wide range of end-users with different interests and priorities; from corporate managers looking for data to improve the performance of their business to software engineers developing intelligent applications for smart environments (see Fig. 1.4).

## 1.4 Book Overview

This book brings together the body of work on Real-time Linked Dataspaces and structures it (as illustrated in Fig. 1.5) into four parts:

- The first part of the book details the motivation and core concepts of Real-time Linked Dataspaces. This part explores the need for an evolution of data management techniques to meet the challenges of data ecosystems for intelligent systems. Chapters in part I cover knowledge sharing among intelligent systems in data ecosystems, fundamentals of the dataspace approach to data management, and introduce the Real-time Linked Dataspace concept and its role as a data platform for intelligent systems within IoT-enabled smart environments.

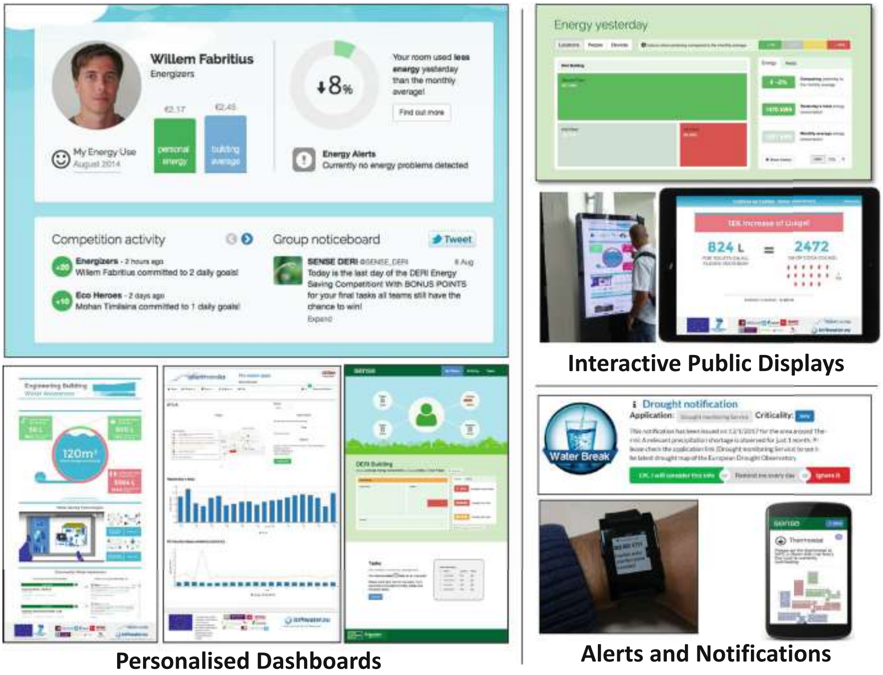


**Fig. 1.4** Intelligent applications built using the Real-time Linked Dataspace [16]

- The second part of the book explores the essential data management support services provided by the Real-time Linked Dataspace. Part II contains chapters that detail data services, including catalog, entity management, query and search, data discovery, and human tasks.
- The third part of the book explores advanced stream and event processing support services for Real-time Linked Dataspaces. Chapters detail advanced techniques for approximate and best-effort stream and event processing services for dataspaces including quality of service, complex event processing, dissemination of IoT streams, and approximate semantic event matching.
- The fourth part of the book explores the use of Real-time Linked Dataspaces within real-world smart environments. The chapters in this part demonstrate the role of the Real-time Linked Dataspace in enabling intelligent water and energy management systems through the development of IoT-based digital twins and intelligent applications, IoT-enhanced user experience, and autonomic source selection for advanced predictive analytics.
- The final part of the book discusses what is required for the widespread adoption of the Real-time Linked Dataspace approach and details a future research agenda for dataspaces, data ecosystems, and intelligent systems.

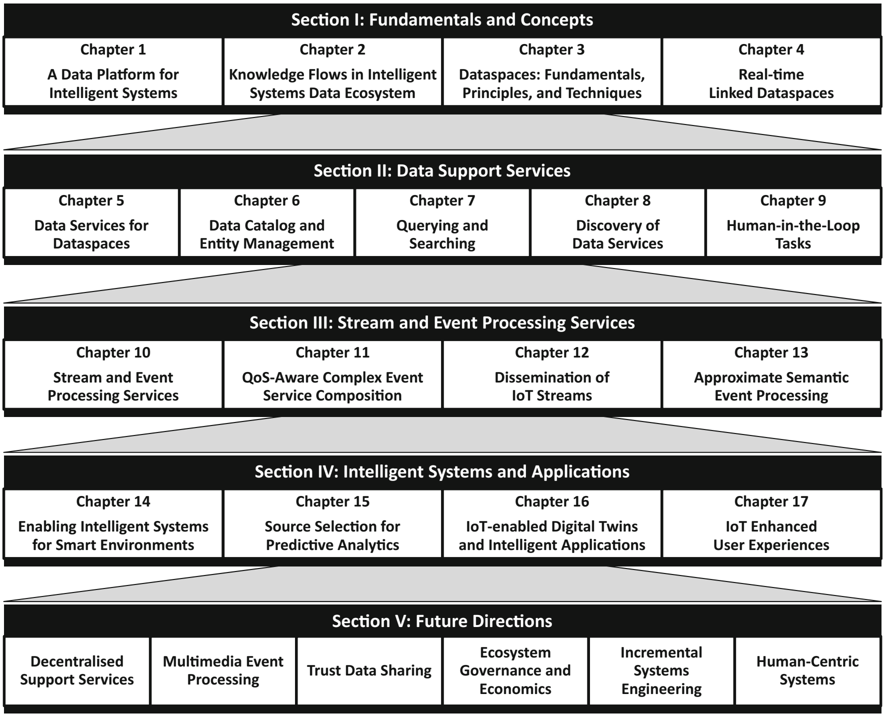

Section I: Fundamentals and Concepts
Chapter 1
A Data Platform for Intelligent Systems
Chapter 2
Knowledge Flows in Intelligent Systems Data Ecosystem
Chapter 3
Dataspaces: Fundamentals, Principles, and Techniques
Chapter 4
Real-time Linked Dataspaces
Section II: Data Support Services
Chapter 5
Data Services for Dataspaces
Chapter 6
Data Catalog and Entity Management
Chapter 7
Querying and Searching
Chapter 8
Discovery of Data Services
Chapter 9
Human-in-the-Loop Tasks
Section III: Stream and Event Processing Services
Chapter 10
Stream and Event Processing Services
Chapter 11
QoS-Aware Complex Event Service Composition
Chapter 12
Dissemination of IoT Streams
Chapter 13
Approximate Semantic Event Processing
Section IV: Intelligent Systems and Applications
Chapter 14
Enabling Intelligent Systems for Smart Environments
Chapter 15
Source Selection for Predictive Analytics
Chapter 16
IoT-enabled Digital Twins and Intelligent Applications
Chapter 17
IoT Enhanced User Experiences
Section V: Future Directions
Decentralised Support Services
Multimedia Event Processing
Trust Data Sharing
Ecosystem Governance and Economics
Incremental Systems Engineering
Human-Centric Systems

**Fig. 1.5** Structure of the book

## 1.5 Summary

In this chapter, we have established the growing importance of intelligent systems as our society undergoes a digital transformation. Internet of Things enabled smart environments will generate vast amounts of data that create new opportunities for intelligent systems. This book postulates the use of the dataspace data management paradigm as the core of a data platform to enable data ecosystems for intelligent systems.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International license (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

cc BY

# Chapter 2
# Enabling Knowledge Flows in an Intelligent Systems Data Ecosystem

**Edward Curry and Adeboyega Ojo**

**Keywords** Intelligent systems · Data ecosystem · Open systems · Digital twins · Internet of Things · Data platforms · Smart environment

## 2.1 Introduction

In data ecosystems, vast amounts of data move among actors within complex information supply chains that can form in different ways around an organisation, community technology platforms, and within or across sectors. This chapter explores the role a data ecosystem can play in the design of intelligent systems to support data-rich Internet of Things (IoT)-based smart environments. The chapter examines different elements of an intelligent systems data ecosystem that are critical to understanding the data management and sharing challenges they present.

In Sect. 2.2, we establish the foundations of an intelligent systems data ecosystem and explore the increasing role data is playing in the design of intelligent systems. Section 2.3 details the challenge to support the exchange of knowledge within open systems in dynamic environments, with Sect. 2.4 outlining the Knowledge Value Ecosystem (KVE) Framework to support knowledge sharing. Sections 2.5, 2.6, and 2.7 explain the framework in more detail and how knowledge, value, and ecosystem barriers are overcome. A pay-as-you-go iterative boundary crossing process to overcome these barriers is discussed in Sect. 2.8. Section 2.9 details the requirements for data platforms to support the sharing of data between intelligent systems within Internet of Things-based smart environments and a summary is provided in Sect. 2.10.

## 2.2 Foundations

As we begin the third decade of the twenty-first century, we are at the beginning of a great wave of convergence of enabling technologies from the Internet of Things (IoT), 5G, high-performance computing, and edge computing to big data, cloud

© The Author(s) 2020
E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_2

computing, and Artificial Intelligence (AI). Smart environments are generating significant quantities of data from digital infrastructure that is driving a new wave of data-driven intelligent systems. Over the last decade, the term "Big Data" was used by different major players to label data with different attributes. The first definition, by Doug Laney of META Group (later acquired by Gartner) defined big data using a three-dimensional perspective: "Big data is high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision-making, insight discovery and process optimization" [20]. Loukides [21] defines big data as "when the size of the data itself becomes part of the problem and traditional techniques for working with data run out of steam." Jacobs [22] describes big data as "data whose size forces us to look beyond the tried-and-true methods that are prevalent at that time". Big data brings together a set of data management challenges for working with data under new scales of size and complexity. Many of these challenges are not new. What is new, however, are the challenges raised by the specific characteristics of big data related to the three V's:

- *Volume (amount of data)*: Dealing with large scales of data within data processing (e.g. healthcare and logistics)
- *Velocity (speed of data)*: Dealing with streams of high-frequency incoming real-time data (e.g. sensors and IoT devices)
- *Variety (range of data types/sources)*: Dealing with data using differing syntactic formats (e.g. spreadsheets, XML, DBMS), schemas, and semantic meanings (e.g. Enterprise Data Integration).

The V's of big data challenge the fundamentals of existing technical approaches and require new forms of data processing to enable enhanced decision-making, insight discovery, and process optimisation. As the big data field matured, other V's have been added, such as Veracity (documenting quality and uncertainty) and Value. The value of data within a smart environment can be considered in the context of the dynamics of knowledge-based organisations [23], where the processes of decision-making and organisational action are dependent on the process of sense-making and knowledge creation.

Through the generation and analysis of data from the smart environment, data-driven systems are transforming our everyday world. From the digitisation of traditional infrastructure (smart energy, water, and mobility), the revolution of industrial sectors (smart autonomous cyber-physical systems, autonomous vehicles, and Industry 4.0), to changes in how our society operates (smart government and cities). At the other end of the scale, we see more human-centric thinking in our systems where users have growing expectations for highly personalised digital services for the "Market of One".

The digital transformation is creating an ecosystem with data on every aspect of our world spread across a range of information systems. Data ecosystems present new challenges to the design of intelligent systems that require a reconsideration of how we deal with the data management needs of large-scale, data-rich smart environments. Intelligent systems need to support openness, flexibility, and dynamicity [24] with the ability to deal with incremental change at minimum cost.

To understand the emerging data management challenges, we explore the design of intelligent systems within smart environments and the need to support knowledge flows within data ecosystems.

### 2.2.1 Intelligent Systems Data Ecosystem

Within a data ecosystem, participants (individual or organisation) can create new value that no single participant could achieve by itself [25]. A data ecosystem can form in different ways, around an organisation, a community of interest (music), a geographical location (city), or within or across industrial sectors (manufacturing, pharmaceutical). In the context of a smart environment, the data ecosystem metaphor is useful to understand the challenges faced with the cross-fertilisation and exchange of knowledge from different intelligent systems within the environment.

A key challenge within the design of intelligent systems is the need to extract valid and accurate insights from the data generated by a smart environment to make useful and meaningful decisions for business and society. Figure 2.1 details the data ecosystem for a connected autonomous vehicle where a community of interacting information systems share and combine their data to provide a holistic functional view of the car, passengers, city mobility, and service and infrastructure providers. Data may be shared about the current operating conditions of the vehicle, traffic flows, or context of the passengers (e.g. a family on holiday or a business executive moving between meetings) to support real-time decision-making, personalised digital services, or data on past observations to improve learning processes.

An intelligent systems data ecosystem (see Fig. 2.2) describes a community of interacting information systems that can share and combine their data to provide a

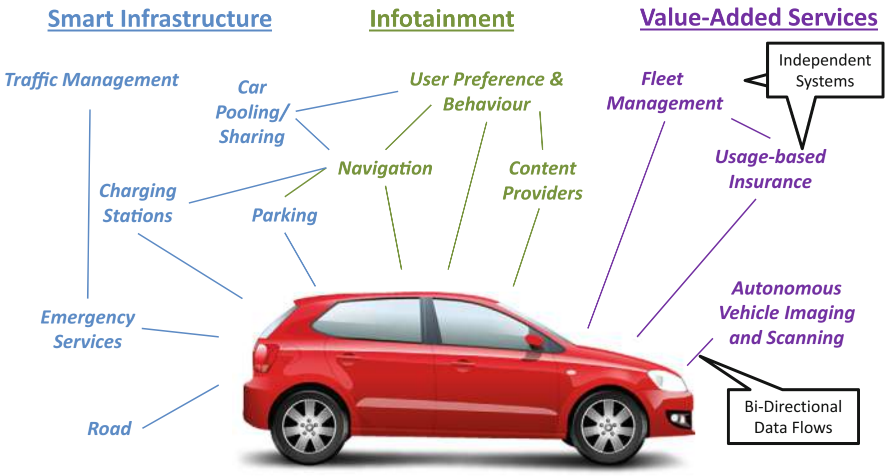


**Fig. 2.1** Connected and autonomous vehicle data ecosystem [1]

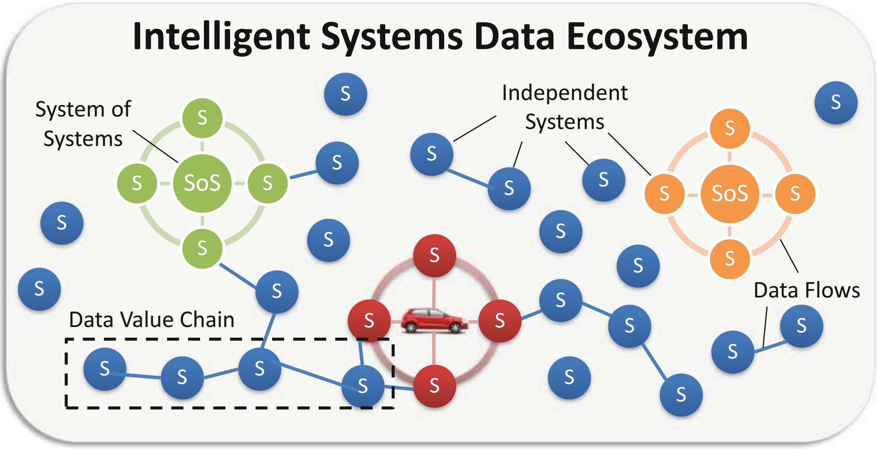


**Fig. 2.2** An intelligent systems data ecosystem

functional view of the environment [1]. The ecosystem supports the flow of data among systems, enabling the creation of data value chains to understand, optimise, and reinvent processes that deliver insight to optimise the overall environment. In a data value chain, information flow is described as a series of steps needed to generate value and useful insights from data [15]. Systems within the ecosystem can also come together to form a System of Systems.

### *2.2.2 System of Systems*

The need for multiple intelligent systems within a smart environment to work together is becoming a standard requirement. Sharing data among intelligent systems is critical if we are to extract the maximum value from IoT-based smart environments. Smart cities are showing how different systems within the city (e.g. energy and transport) can collaborate to maximise the potential to optimise overall city operations [26]. Digital services are expected to deliver a personalised and seamless user experience by bringing together relevant user data from multiple systems [16]. Building these systems requires a System of Systems (SoS) approach to connect systems that cross organisational boundaries, come from various domains, (e.g. finance, manufacturing, facilities, IT, water, traffic, and waste) and operate at different levels (e.g. region, district, neighbourhood, building, business function, individual). The joint ISO/IEC/IEEE definition of an SoS is that it “brings together a set of systems for a task that none of the systems can accomplish on its own. Each constituent system keeps its management, goals, and resources while coordinating within the SoS and adapting to meet SoS goals” [27]. Maier [28] identified a set of characteristics to describe an SoS:

- *Operational Independence*: Constituent systems can operate independently from the SoS and other systems.
- *Managerial Independence*: Different entities manage the constituent systems.
- *Geographic Distribution*: Is the degree to which a system is widely spread or localised.
- *Evolutionary Development*: The SoS, and its behaviour evolves, requiring changes to system interfaces to be maintained and kept consistent.
- *Emergent Behaviour*: New emergent behaviour can be observed when the SoS changes.

There are many systems engineering challenges in bringing together the constituent systems into an SoS at the data, service, process, and organisational levels. Many of the above characteristics of an SoS give us insights into the knowledge, value, and ecosystem boundaries that exist in bringing an SoS together, and the different types of interests possible at management and operational levels of systems. At the data level, intelligent systems can benefit from leveraging data from the availability of large volumes and variety of data and streams in the smart environment, which can be used to fuel intelligent, evidence-based decision-making.

### 2.2.3 From Deterministic to Probabilistic Decisions in Intelligent Systems

When it comes to making decisions in intelligent systems, there are two general approaches: deterministic (model-driven) and probabilistic (data-driven). A critical difference between the approaches can be explored by considering the costs and level of reliability and adaptability they provide within intelligent systems. There is a tension between reliability, predictability, and cost [29]: usually the more dependable and reliable the intelligent system needs to be, the more cost is associated with its development. Typically, we can see deterministic systems as reliable but with high costs to develop and adapt, and probabilistic as low cost to build and adapt, but less reliable. Take the example of the autonomous connected car, where we have the strict requirements of safety-critical autonomous driving systems (where a failure may lead to loss of life or serious personal injury) to the "good enough" requirements of the infotainment systems (where a failure is acceptable and merely an inconvenience to the user).

Within early smart environments, the level of data available was limited due to the high cost of digitisation. Sensors were expensive to purchase and install, resulting in the prudent use of resources. Conventional intelligent systems typically targeted "high-value" opportunities where the cost savings and benefits could justify the high cost of investment needed. Often these would be safety- or mission-critical systems that required higher levels of reliability. Due to the lack of sensor data and the need for high levels of reliability, deterministic approaches were an obvious choice for "conventional" intelligent systems. In this approach, the environment is optimised based on a formal deterministic model where a set of rules and/or equations detail the decision logic for the intelligent system that is used to control the activity in the environment efficiently and predictably. Adapting the intelligent system to meet changes in the environment is a costly process as the model, and its rules, need to be updated by expert systems engineers.

In the probabilistic approach, the core of the decision process is a statistical model that has been learnt from an analysis of "training" data to "discover" the structure of a decision model automatically from the observed data (e.g. driver behaviour). Thus, a fundamental requirement of data-driven approaches is the need for data to fuel the training of the algorithms. A lack of data, and training data, within a smart environment has limited the use of data-driven approaches.

As the IoT is enabling the deployment of lower-cost sensors, we are seeing more extensive adoption of IoT devices/sensors and gaining more visibility (and data) into smart environments. Smart environments are generating different types of data with an increase in the number of multimedia devices deployed, such as vehicle and traffic cameras. The emergence of the Internet of Multimedia Things (IoMT) is resulting in large quantities of high-volume and high-velocity multimedia event streams that need to be processed [30]. The result is a data-rich ecosystem of structured and unstructured data (e.g. images, video, audio, and text) detailing the smart environment that can be exploited by data-driven techniques. It is estimated that a single connected car will upload about 25 gigabytes of data per hour, while a vehicle fitted with an Autonomous Vehicle Imaging and Scanning system generates and processes about 4 TB of data for every autonomous driving hour (https://www.datamakespossible.com/evolution-autonomous-vehicle-ecosystem/).

The increased availability of data has opened the door for the use of data-driven probabilistic models, and their use within smart environments is becoming more and more commonplace for "good enough" scenarios. As a result, the conventional rule-based approach is now being augmented with data-driven approaches that support optimisations driven by machine learning, cognitive and AI techniques that are opening new possibilities for the design of intelligent systems. For example, pedestrian detection is challenging to implement in a rule-based approach. However, deep learning models for object detection and semantic segmentation using a dash-mounted camera are highly effective at detecting pedestrians.

Intelligent systems can now adapt to changes in the environment by leveraging the data generated in the environment within their learning process to improve performance. If intelligent systems share data on their operational experiences, a pool of data can be created to improve the overall learning processes of all the systems, a form of collective AI through the "wisdom of the systems". Because the process is data-driven, it can be run and re-run at low cost. This critical role of data in enabling adaptability and collective machine intelligence makes it a valuable resource.

Within the context of smart environments, data-driven approaches have been used to optimise the operation of infrastructure, such as the energy and transportation systems [31]. However, the adoption of data-driven approaches is about to increase significantly across a range of industries and sectors with the use of Digital Twins.

### 2.2.4 Digital Twins

Within the business community, the metaphor of a "Digital Twin" is gaining popularity as a way to explain the potential of IoT-enabled assets and smart environments [32]. A digital twin refers to a digital replica of a physical asset (car), processes (value chain), system (transport), or physical environment (building). As illustrated in Fig. 2.3, the digital twin provides a digital representation (i.e. simulation model or data-driven model) that updates and changes as the "physical twin" changes. The digital representation provided by the digital twin can be analysed to optimise the operation of the physical twin.

Digital twins are constructed from multiple sources of data, including real-time IoT sensors, historical sensor data, traditional information systems, and human input from domain and industrial experts. With the use of advanced analytics, machine learning, and AI techniques, the digital twin can learn the optimal operating conditions of the physical twin and optimise the physical twins' operations in areas such as performance, maintenance, and user experience. One of the most promising outputs from a digital twin analysis is the possibility to find root causes of potential anomalies which can happen (prediction) and improve the physical process (innovation).

Digital twins can range from human organs such as the heart and lungs to aircraft engines and city-scale twins. For example, the SmartSantander smart city project has deployed tens of thousands of IoT-connected sensor devices in large cities across

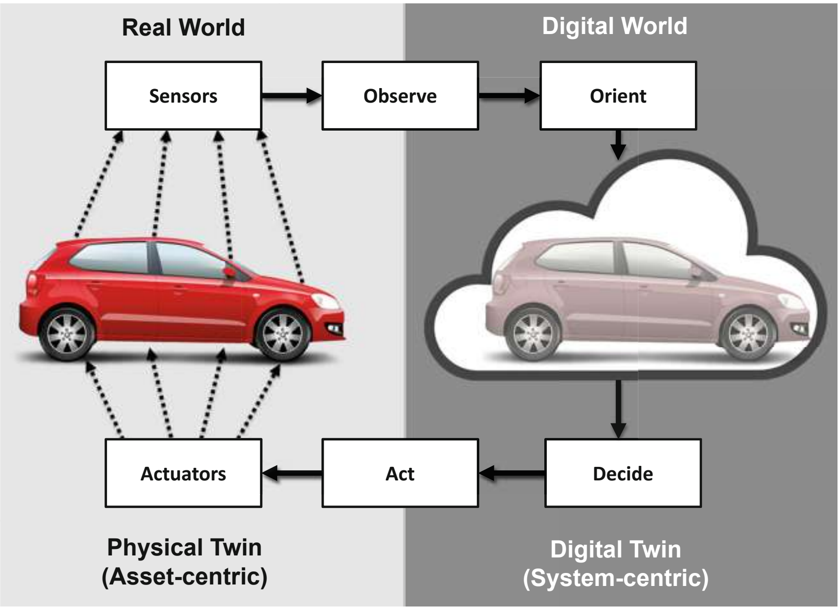


**Fig. 2.3** Information flow and processing steps within a digital twin

Europe [33]. The sensing capabilities of these devices are wide-ranging, including solar radiation, wind speed and direction, temperature, water flow, noise, traffic, public transport, rainfall, and parking. The devices provide a digital representation of the city, which enables visibility into city processes and operations to support analysis and optimisation.

Datafication is creating an ecosystem of data on every aspect of our world spread across a range of information systems. In order for digital twins and intelligent systems to maximise the benefits from the resulting data ecosystems, we need to rethink how we exchange knowledge among open intelligent systems in dynamic environments.

## 2.3 Knowledge Exchange Between Open Intelligent Systems in Dynamic Environments

The design of intelligent systems, especially ones enabled by IoT, has to accommodate the needs of dynamic environments, where system participants continuously join and leave the environment. Vermesan et al. call this phenomenon "fluid systems" that are continuously changing and adapting, "in IoT systems it is very common to have nodes that join and leave the network spontaneously" [34]. This dynamic nature puts constraints on the assumptions that can be made within the design of intelligent systems and the assumption of having full understanding or control over the systems in the environment. This has led to the need for "open" intelligent systems which can adapt to their environment and learn from its interaction with the changing dynamics of the environment and the different systems operating within it.

While the term "open" has been frequently used in the literature to describe large-scale distributed systems, for example, Ciliaet et al. [35], a broad consensus has not been reached on its definition. Looking to the early works in system theory, we can draw upon the definition commonly used in this field as a system that has external interactions in the form of information, energy, or matter transfer through the system boundary [36]. A boundary here separates the system from its environment. For example, in biology, a cell exchanges chemicals with its environment through its membrane, and thus, it is an open system from this perspective.

The concept of boundary objects is established in the literature on knowledge sharing and reuse. Boundary objects are used to understand and coordinate the interactions among actors with varying information and knowledge needs to establish a shared point of reference during interactions [37]. Carlile formulates suggestions for managing knowledge across boundaries and provides the 3-T framework for knowledge exchange across system boundaries within the area of organisation science [37]. While Carlile's framework focuses on the exchange of knowledge between product development teams (the "systems" in this case), its foundations can be traced back to the Shannon-Weaver model with implications for information systems. The 3-T framework defined the task of knowledge exchange as a task of crossing three boundaries among systems: syntactic, semantic, and pragmatic. We interpret these boundaries from Carlile for knowledge exchange among open systems in dynamic environments and extend them with new boundaries for value exchange and ecosystem coordination.

In Fig. 2.4, the inverted pyramidal frustum shape shows the spectrum of tasks between well-known and novel tasks that need to be undertaken to exchange knowledge within an ecosystem. Systems A and B interact within this spectrum with correspondence to boundary objects that exist at three levels:

- *Knowledge Boundaries*: Exist where differences and dependencies among systems exist at the semantic and the administrative levels. A common lexicon needs to be developed to transfer and assess knowledge among systems in the classical sense from Shannon [38]. However, as Shannon noted, a common lexicon may not always be sufficient to share knowledge among systems. Distinct systems will have differences and dependencies that are unclear with multiple possible interpretations which create a semantic boundary to knowledge sharing. The administrative boundary describes how close or far in terms of control are the systems. A close control means that many assumptions can hold concerning data management guarantees (e.g. data consistency, availability, and quality), while a far control refers to weaker or no guarantees. To cross the knowledge boundary, it is necessary to develop common meanings to provide a means of sharing and assessing knowledge at a boundary. This requires new agreements on the translation of each system to the commonly shared meaning and an agreed upon protocol for access.
- *Value Boundaries*: Systems generally serve the interest of their participants, with different systems serving the different interests of their users. Cultural, organisational, and social interests can impede the sharing of knowledge among systems. To overcome the value boundary, it is necessary to develop common

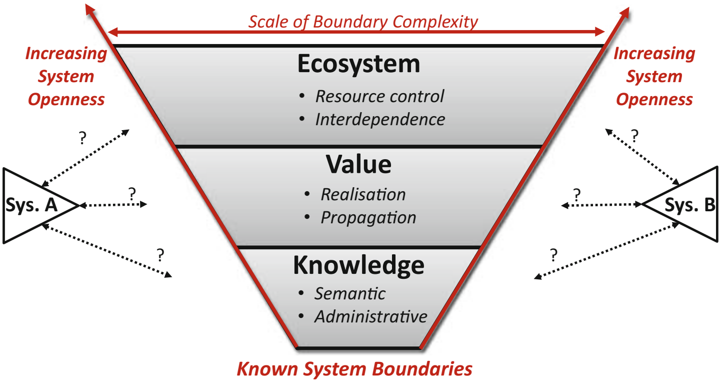


**Fig. 2.4** Knowledge exchange between two systems within an ecosystem. Based on concepts from the 3-T Framework [37]

interests among systems, and their participants, to provide sufficient motivation for knowledge sharing. The value transformation necessary to create common interests requires significant practical and political effort, and the value must be propagated within the ecosystem.

- *Ecosystem Boundaries*: Ecosystems generally have different levels of interdependence between systems in both the technical and management sense. The ecosystem can create the conditions for a marketplace competition among participants or enable collaboration among diverse, interconnected participants that depend on each other for their mutual benefit. Another key factor is the control of key data resources within the ecosystem. Who own the key data resources? Is the data available to all participants in the ecosystem? Are there commercial terms of use? A close ecosystem coordination framework would provide clear answers to these questions, while loose coordination means less predictability on the behaviour of participants within the data ecosystem. To overcome ecosystem boundaries, it is necessary to understand and support the social, political, organisational, and business changes needed for ecosystem coordination.

The more open, distributed, and heterogeneous the environment becomes, the more significant these boundaries become, especially the latter ones where openness may introduce more novelty and uncertainty. Crossing boundaries requires mutual agreements among participants, which implies cost. The need for mutual agreements among participants adds to the technical issues an essential social dimension. Overcoming the differences among systems generates costs to the systems involved where domain-specific knowledge, as well as the common knowledge used, may need to be transformed to share and assess knowledge among the systems effectively.

There is an inherent need to design intelligent systems with the ability to scale and cross system boundaries. To effectively cross the ecosystem boundary for multiple systems within an open environment, each system must be able to represent current and more novel forms of knowledge, learn about their consequences, and transform their domain-specific knowledge accordingly. Intelligent systems within dynamic environments need to support the "social" agreement needed to share knowledge among them. We capture the capabilities needed to overcome knowledge sharing barriers among intelligent systems in the KVE Framework.

## 2.4 Knowledge Value Ecosystem (KVE) Framework

In order to cross the three boundaries of sharing knowledge among open intelligent systems in dynamic environments, we propose the Knowledge Value Ecosystem (KVE) Framework (Fig. 2.5). The KVE Framework, an extension of the 3-Ts Framework, tackles each boundary using the following capabilities:

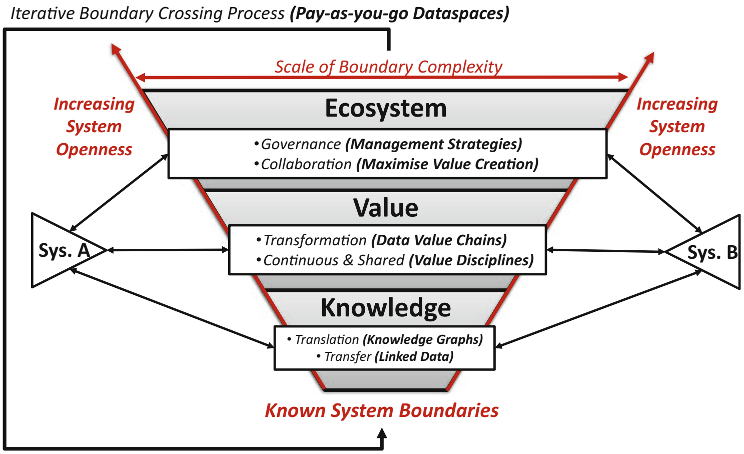

Iterative Boundary Crossing Process (Pay-as-you-go Dataspaces)
Scale of Boundary Complexity
Increasing System Openness
Increasing System Openness
Ecosystem
•Governance (Management Strategies)
•Collaboration (Maximise Value Creation)
Value
•Transformation (Data Value Chains)
•Continuous & Shared (Value Disciplines)
Knowledge
• Translation (Knowledge Graphs)
• Transfer (Linked Data)
Sys. A
Sys. B
Known System Boundaries

**Fig. 2.5** The Knowledge Value Ecosystem (KVE) Framework for enabling knowledge flows within an ecosystem

- *Knowledge Transfer and Translation*: Knowledge boundaries are crossed by tackling administrative barriers using capabilities for the transfer of data using open web standards for data publishing and sharing knowledge as linked data, while semantic boundaries are crossed with capabilities to establishing common meanings among intelligent systems using knowledge graphs (see Sect. 2.5.3). Together, linked data and knowledge graphs can be used to support an incremental approach to reaching agreements on the transfer and translation of meaning among multiple intelligent systems.
- *Continuous and Shared Value Transformation*: Value boundaries are crossed with capabilities for transforming the interests of individual participants into common interests within new shared data value chains. The shared data value chain approach can provide a clear value proposition to support the political effort necessary from both a business case and an organisational perspective. It is important that the value created is continuously shared between participants along the value chain to motivate their contribution and support the sustainability of the data ecosystem.
- *Ecosystem Governance and Collaboration*: The nature of the ecosystem, the participants, and their dynamics will affect the management strategies needed to support the social, political, and organisational changes needed. Within a well-functioning data ecosystem, the participants are efficiently and effectively collaborating to exchange knowledge to maximise value creation.
- *Iterative Pay-As-You-Go Process*: Typically, the process of crossing boundaries, especially ecosystem boundaries, cannot be resolved with a single attempt. It requires an *Iterative Boundary Crossing Process* which supports trial and error in transforming complex knowledge across system boundaries. An iterative approach can support a learning process to improve the boundary crossing

**Table 2.1** Boundaries, barriers, and capabilities within the KVE Framework and proposed implementation within a data platform

| Boundary | Barrier | Capability | Data platform implementation |
|---|---|---|---|
| Knowledge | Administrative | Transfer | Linked data |
| | Semantic | Translation | Knowledge graphs |
| Value | Realisation | Transformation | Data value chains |
| | Propagation | Continuous and shared | Value disciplines for data networks |
| Ecosystem | Resource control | Governance | Management strategies |
| | Interdependence | Collaboration | Maximise value creation |
| | | Iterative boundary process | Pay-as-you-go dataspaces |

capability. Dataspaces provide a pay-as-you-go approach to support incremental data management within the ecosystem.

Within this approach, there is natural support between the different capabilities as linked data can support the definition and sharing of knowledge graphs, which can then both support the creation of value chains in a data ecosystem and motivate the need for collaboration between participants. The first capabilities (knowledge transfer and translation) can be captured in the technical design of a data platform. The second (continuous and shared value transformation) requires a higher-level value transformation among systems together with a cultural transformation of the stakeholders to promote data sharing and creation of new data value chains among systems. The purpose of the third capability (governance and collaboration) is to gradually improve the overall operation of the ecosystem to maximise benefits for all participants. Finally, the iterative boundary crossing process can support all the capabilities to improve over time following a pay-as-you-go approach. The alignment of the boundaries, barriers, capabilities, and their implementation is detailed in Table 2.1. We will now introduce each of these capabilities in more detail and explore how they can be implemented within the design of a data platform to support knowledge sharing among intelligent systems in a data ecosystem.

## 2.5 Knowledge: Transfer and Translation

In order to cross the knowledge boundaries of systems, two capabilities are needed: transfer and translation. Within the KVE Framework, knowledge boundaries are crossed by using an entity-centric model that establishes common meanings among systems using knowledge graphs expressed using linked data.

### 2.5.1 Entity-Centric Data Integration

Data integration projects typically focus on one-off point-to-point integration solutions among two or more systems in a customised but inflexible and non-reusable manner—this limits both the information flow and its oversight among systems to those that have been integrated. Entity-centric data integration takes a different form to traditional schema-level integration within the relational model. The entity-centric data integration model is based on global identifiers representing objects or concepts that can be reused or reconciled among different datasets (or systems).

Entity-centric data integration facilitates the co-existence of different perspectives and points of view of entities and a decentralised evolution of the data. At the same time, the use of linked data vocabularies, and the specification of conceptual models for a domain under the Resource Description Framework (RDF) model are used to facilitate the interoperability and semantic integration among different datasets for specific domains. This entity-centric integration of knowledge graphs using linked data has a number of virtues to represent large, complex, and heterogeneous conceptual models as detailed by [39–41]:

- *Support for the representation of sparse data*: RDF(S) is based on a graph data model, which supports a sparse data model.
- *Schema flexibility*: RDF(S) datasets are schema-less and can evolve in a decentralised manner.
- *Represent and map to/from other data models*: Data in a relational or in other formats (e.g. CSV) can be represented and systematically mapped to RDF [42].

These characteristics make entity-centric knowledge graphs an ideal approach for establishing a shared meaning among systems to cross knowledge boundaries. When knowledge graphs are expressed using linked data, they can be created in a fashion that allows two systems to be easily linked to each other on the information-level (data) not the infrastructure-level (system) by focusing more on the conceptual similarities (shared understanding). The combination of knowledge graphs and linked data meets many of the FAIR data principles for data management (see Sect. 2.8), including persistent identifiers, metadata, and open protocols. The approach provides a means for translating knowledge across the knowledge boundaries among systems. It allows separate systems that were designed independently to be later joined/linked at the edges, for interoperability to be added incrementally when needed and where cost-effective, and for the meaning of data to be expressed in a mixture of vocabularies.

### 2.5.2 Linked Data

In order to cross the administrative boundaries of systems to support data transfer, we propose the use of linked data. Linked data leverages open protocols and W3C standards of the web architecture for sharing structured data on the web. The fundamental concept of linked data is that data is created with the mindset that it will be shared and reused by others. The objective is to expose the data within existing systems but only link the data when it needs to be shared. Linked data provides a decentralised incremental approach for information sharing based on the creation of a global information space [43]. Linked data has the following characteristics:

- *Open*: Linked data is accessible through a variety of applications because it is expressed in open, non-proprietary syntactic format.
- *Modular*: Linked data can be combined (mashed-up) with any other pieces of linked data. No planning is required to integrate two data sources if they both use linked data standards.
- *Scalable*: It is easy to add and connect more linked data to existing linked data, even when the terms and definitions that are used change over time.

Linked data uses standards, tools, and techniques from work on the semantic web to facilitate sharing and reuse of data across domains. It primarily uses a graph-based representation framework for structuring data and uses standard ontology languages for defining the semantics of data. Ontologies (or vocabularies) provide a shared understanding of concepts and entities within a domain of knowledge which supports automated processing for data using semantic web tools. Thus, the use of linked data at the syntactic level can support the establishment of a common lexicon. At the semantic level, it can also support the establishment of shared meanings.

Linked data when used together with the dataspace approach provides a framework for a decentralised pay-as-you-go data integration with a standardised data model representation providing a minimum level of integration and where Universal Resource Identifiers (URIs) and the Domain Name Systems (DNS) provide a global-level identification scheme, which facilitates the referencing of data entities among different datasets. The RDF standard provides a common interoperable format and model for data linking and sharing on the web. RDF is the basic machine-readable representational format used to represent information. It is a general method for encoding graph-based data that is self-describing, meaning that the labels of the graph describe the data itself.

Linked data uses web standards in conjunction with four basic principles for exposing, sharing, and connecting data. These principles are:

- *Naming*: Use of URIs as names to identify things such as a person, a building, a device, an organisation, an event or even concepts such as risk exposure or energy and water consumption, simplifies reuse and the integration of data.
- *Access*: Use of URIs based on HyperText Transfer Protocol (HTTP) so that people can look up those names—URIs are used to retrieve data about objects using standard web protocols. For an employee, this could be their organisation and job classification, for an event, this may be its location time and attendance, for a device, this may be its specification, availability, and price.

- *Format*: When a URI is looked up (dereferenced) to retrieve data, it provides useful information using a standardised format, ideally, in web standard formats such as RDF.
- *Contextualisation*: Include links to other URIs so that more information can be discovered. Retrieved data may link to other data, thus creating a data network; for example, data about a product may link to all the components it is made of, which may link to their supplier.

Using these technologies, we can support data transfer among intelligent systems by using: (1) URIs to name things; (2) RDF for representing data; (3) Linked data principles for publishing, linking, and integration; (4) Vocabularies to establish and share understanding; and (5) Bottom-up incremental agreement.

### *2.5.3 Knowledge Graphs*

Overcoming semantic boundaries among systems requires a common understanding of meaning among systems for knowledge to be shared. Within the KVE Framework, semantic boundaries are crossed by establishing common meanings among systems using knowledge graphs expressed using linked data. Knowledge graphs and linked data can be used to support an incremental approach to reaching agreements on the translation of the meaning of knowledge among systems.

In 2012 Google coined the term "Knowledge Graph" to refer to their use of information gathered from multiple sources to enrich their services, including search engine results. The term has also been used to refer to Semantic Web knowledge bases such as DBpedia or YAGO. As defined by Paulheim [44] a "knowledge graph (1) mainly describes real-world entities and their interrelations, organised in a graph, (2) defines possible classes and relations of entities in a schema, (3) allows for potentially interrelating arbitrary entities with each other and (4) covers various topical domains." As illustrated in Fig. 2.6, a knowledge graph is just a set of entities (e.g. Marie Curie and France), a set of relations between those entities' (e.g. "knownFor" and "wasResidentOf"), and a set of facts (see Table 2.2). Facts are the combination of the entities and relationships "Marie Curie, wasResidentOf, France". More formally, a knowledge graph is a tuple (E, R, G), where:

- E is a set of nodes, each representing an **entity** in the domain.
- R is a set of edge labels, each representing a **predicate**, or a semantic relation type.
- $G \subseteq E \times R \times E$ is a set of $\langle$subject, predicate, object$\rangle$ **triples**, denoting **facts**.

Knowledge graphs provide a flexible knowledge representation structure that can describe entities and concepts that may come from multiple systems and domains, and at varying levels of granularity. Knowledge graphs can be used to create large knowledge bases (see Table 2.3). However, managing graphs of these sizes poses several challenges regarding quality, coherence, performance, and interaction.

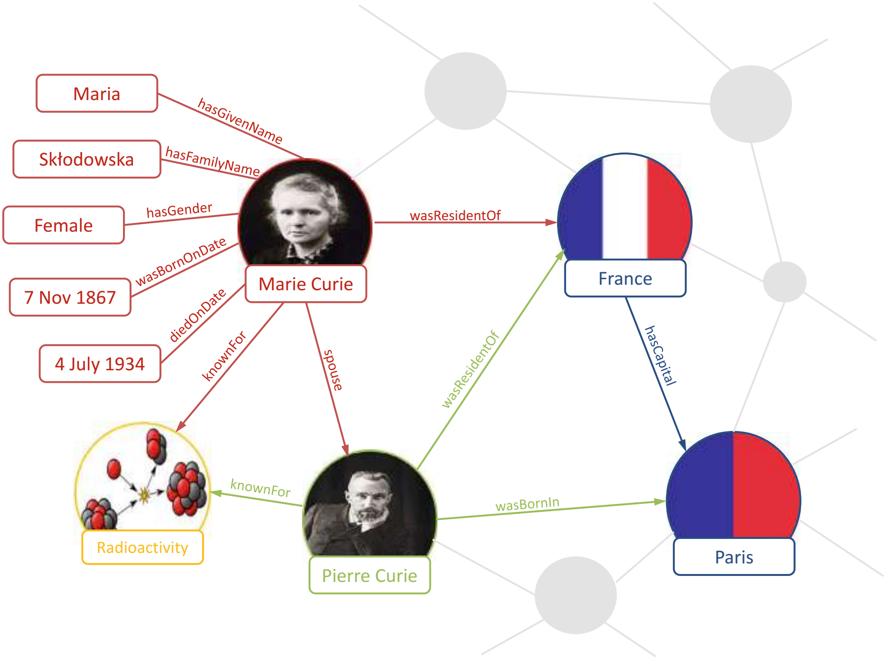


**Fig. 2.6** Example of a knowledge graph for Marie Curie

**Table 2.2** Facts of the knowledge graphs for Marie Curie

| Subject | Predicate | Object |
|---|---|---|
| Marie Curie | hasGivenName | Maria |
| Marie Curie | hasFamilyName | Skłodowska |
| Marie Curie | hasGender | Female |
| Marie Curie | Spouse | Pierre Curie |
| Marie Curie | knownFor | Radioactivity |
| Marie Curie | wasBornOnDate | 7 Nov 1867 |
| Marie Curie | wasResidentOf | France |
| Marie Curie | diedOnDate | 4 July 1934 |
| France | hasCapital | Paris |

**Table 2.3** Size of some schema-based knowledge bases [45]

| Knowledge graph | #of Entities | # of Relation types | # of Facts |
|---|---|---|---|
| Freebase | 40 M | 35,000 | 637 M |
| Wikidata | 18 M | 1632 | 66 M |
| DBpedia (en) | 4.6 M | 1367 | 538 M |
| YAGO2 | 9.8 M | 114 | 447 M |
| Google Knowledge Graph | 570 M | 35,000 | 18,000 M |

### 2.5.4 Smart Environment Example

An example of an entity-centric knowledge graph expressed as linked data within the context of a smart environment is illustrated in Fig. 2.7. Data and facts are specified as statements and are expressed as atomic constructs of a subject, predicate, and object, also known as a triple. The statement "Main Kitchen contains Coffee Machine" is expressed in the triple format as:

Subject—"Main Kitchen"
Predicate—"contains"
Object—"Coffee Machine"

RDF is designed for use in web-scale decentralised knowledge graph data models. For this reason, the statement parts need to be identified so that they can be readily and easily reused. RDF uses URIs for identification, so by expressing the previous statement in RDF it becomes:

http://data.deri.ie/rooms#r315
http://vocab.deri.ie/rooms#contains
http://water.deri.ie/devices#mr-coffee

URIs that describe the data can be uniformly used across systems, even if they come from different sources. The knowledge graph structure of the linked data, as illustrated in Fig. 2.7, easily supports optional parameters, and the evolution of parts of the data structure does not affect any other related data. The relations are described on a low-level; therefore, they combine (linking) pieces of data based on their relation types, and not only on their representation.

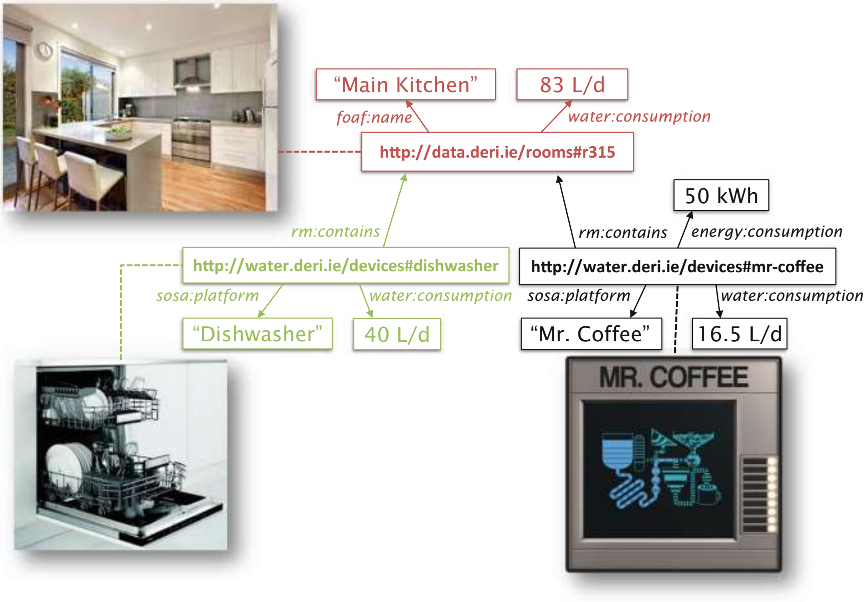


Fig. 2.7 Example of data linkage using URIs and RDF vocabularies in a smart environment

The flexibility to represent data and to support different relationships is a key benefit of linked data to support the sharing of data among systems. Linked data's use of vocabularies and ontologies is an important tool to establish shared meanings among different systems incrementally. This capability is critical to cross knowledge boundaries among systems with the use of knowledge graphs and entity-centric data integration to support the translation of knowledge.

## 2.6 Value: Continuous and Shared

The next part of the KVE Framework tackles value boundaries by identifying the common interests (data value chains) needed to support a value transformation for systems to share knowledge. We explore value disciplines and data network effects and how they can create new opportunities for the participants within the data ecosystem. These value opportunities can be the source of common interest to motivate the social, cultural, or business transformation needed to support knowledge exchange.

### *2.6.1 Value Disciplines*

A value proposition is shaped by an underlying value discipline which describes different ways an organisation or system can differentiate itself from competitors. A strong value proposition can set the strategic focus that enables organisations or systems to set its vision and objectives. It can then tailor its value disciplines to match the need exactly. Treacy and Fred Wiersema [46] created a model to describe three generic value disciplines: (1) Operational Excellence, (2) Product Leadership, and (3) Customer Intimacy. The use of value disciplines has been explored in the broader areas of digital value [47], but also more specific areas such as open data [48]. Within the context of this work, we explore the use of data value disciplines to understand the value opportunities that are possible from data within a data ecosystem. The value of data within a smart environment can be considered in the context of the dynamics of knowledge-based organisations [23], where the processes of decision-making and organisational action are dependent on the process of sense-making and knowledge creation. Based on existing work [46–48], we identify the following three value disciplines for the participants (e.g. user, system, or organisation) of an intelligent systems data ecosystem:

- *Utility*: Tailors the value proposition to directly support the information needs of the participants. The objectives and information requirements of the participants should be defined to determine the usefulness of the data shared within the ecosystem. The utility can be shared between or can be unique to each participant.
- *Performance*: Tailors the value proposition to match to the needs of the participants specifically for improving processes for operational excellence. This can

result in greater efficiencies with associated cost avoidance. Participants with this orientation aim to share data which support their primary performance objectives.

- *User Intimacy*: Tailors the value proposition to directly support the needs of users within the environment by providing information to enhance personalised user experiences and services.

### 2.6.2 Data Network Effects

A network effect is a positive effect described in economics and business where an additional user of a product or service has a positive effect on the value of that product/service to others. When a network effect is present, the value of the product or service increases according to the number of others using it. Robert Metcalfe popularised network effects (also called network externality or demand-side economies of scale) within the context of Ethernet as Metcalfe's law [49]. Within the area of data, network effects are starting to emerge, although in different forms, at both the data ecosystem and data product/service levels.

At the ecosystem level, the network effect can be seen as more systems/users join and contribute data to the data ecosystem; the overall data ecosystem becomes more valuable for the different value disciplines, see Fig. 2.8. Initiatives such as smart cities are showing how different sectors (e.g. energy and transport) can share data to maximise the potential for optimisation and value return. Data network effects occur at the data product/service level, where the data product/service becomes smarter (e.g. predictions, recommendations, and personalisation) as it gets more data from other participants. Leveraging data network effects requires a learning process within the data produce/service that uses advanced analytics to extract insights from the collected data. The data network effect from cross-fertilisation of stakeholder and datasets from different sectors is a crucial element for advancing the big data economy in Europe [15] and is critical to support the value proposition of data ecosystems to their participants.

| **Source of Data** | Utility | Performance | User Intimacy |
|---|---|---|---|
| Ecosystem (Data Network Effects) | *Holistic and long-tail insight across systems* | *Global optimisations* | *Holistic personalised user journey across systems* |
| Single System | *System-level insight* | *Local optimisations* | *Personalised user journey* |

**Value Discipline**

**Fig. 2.8** Value transformation opportunities across value disciplines at the single system and ecosystem levels

## 2.7 Ecosystem: Governance and Collaboration

In order to understand some of the "political" and organisational issues that occur at ecosystem boundaries among systems, this section examines the work on business ecosystems to see how governance and collaboration can support knowledge flows. The section discusses data ecosystem coordination and the range of possible ecosystem design options.

### *2.7.1 From Ecology and Business to Data*

The term Ecosystem was coined by Tansley in 1935 [50] to identify a basic ecological unit comprising of both the environment and the organisms that use it. Within the context of business, James F Moore [51–53] exploited the biological metaphor and used the term to describe the business environment. Moore defined a business ecosystem as an "economic community supported by a foundation of interacting organisations and individuals" [53]. A strategy involving a company attempting to succeed alone has proven to be limited regarding its capacity to create valuable products or services. It is crucial that businesses collaborate among themselves to survive within a business ecosystem [52, 54]. Innovation Ecosystems allow companies to create new value that no company could achieve by itself [55]; often, the ecosystem is centred around the technology platform or technology leadership of a focal firm.

The business ecosystem perspective is a more holistic way to look at the benefits of collaboration among companies, or in the case of a smart environment, the benefits of collaboration among systems. The ecosystem metaphor is, again, a useful metaphor to describe the data within and surrounding a smart environment. A data ecosystem is a socio-technical system enabling value to be extracted from data value chains supported by interacting organisations and individuals [15]. Data ecosystems can form in different ways around organisations, communities, technology platforms, or within or across sectors. Data ecosystems exist within many industrial sectors where vast amounts of data move among actors within complex information supply chains. Sectors with established or emerging data ecosystems include healthcare, finance [56], logistics, media, manufacturing, and pharmaceuticals.

In natural ecosystems, smart organisms control their energy. In business ecosystems, a smart company manages information and its flows [57]. In data ecosystems, a smart company extracts the maximum value from the available data. The ecosystem can create the conditions for a marketplace competition among participants or enable collaboration among diverse, interconnected participants that depend on each other for their mutual benefit. Data ecosystems are useful for creating common interests among systems that are needed for the value transformation required to share data. The benefits of sharing and linking data across domains and industry sectors are becoming apparent with the potential for new value opportunities on the Web of Data.

### *2.7.2 The Web of Data: A Global Data Ecosystem*

The web is moving from a medium for sharing documents to a medium that can also be used to share data. Fuelled by the *Open Data* initiative, the emerging "Web of Data" means easier access to data for users. Typically, open data is non-textual material such as maps, genomes, chemical compounds, mathematical, and scientific formulae. Open data can also include generalised business news, product information, and financial data [56] available from an assortment of sources. Demands for higher levels of transparency have resulted in Open Government initiatives that have made available large numbers of statistical, financial, and economic datasets for public consumption. A number of large-scale knowledge bases have been made available from both private and not-for-profit initiatives, including Google Knowledge Graph, DBpedia, and YAGO, to name a few. The LinkedIn Economic Graph describes all the data on LinkedIn like companies, members, and jobs, to provide a digital representation of the global economic activity with a focus on employment opportunities. The Linked Open Data Cloud represents a large number of interlinked RDF datasets within the broader ecosystem that is being actively used by industry, government, and scientific communities [58]. The linked data cloud has been growing in the past years and provides a foundation upon which applications can be built. The Facebook Open Graph describes a rich object in a social graph, simplifying the process of sharing social data on the web. The Schema.org imitative was founded by Google, Microsoft, Yahoo, and Yandex to create shared vocabularies through an open community process for publishing data. Schema.org vocabularies can be used with many different encodings, including RDFa, Microdata, and JSON-LD. These vocabularies cover entities, relations among entities and actions, and can easily be extended through a well-documented extension model. Each of these initiatives is part of a broader data ecosystem in the emerging Web of Data.

### *2.7.3 Ecosystem Coordination*

Within the KVE Framework, the role of the data ecosystem and data value chains is to support the value transformations necessary (social, cultural, value) to create new common interests and value opportunities among intelligent systems. To achieve this, it is necessary to understand and support the social, political, and organisational changes needed for coordinating ecosystems. To understand the dynamics of an intelligent systems data ecosystem we can look into the literature on System of Systems [28] and Business Ecosystems [59] to enable us to understand the data ecosystems that can exist [1]. In Fig. 2.9, we can see the different types of data ecosystems that can form around intelligent systems within a smart environment. Two critical criteria that influence the design of a data ecosystem and the relationships among participants are:

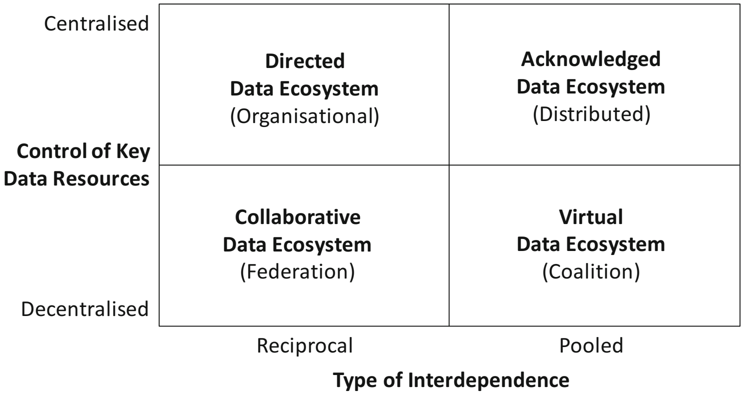

Centralised
Control of Key Data Resources
Decentralised
Directed Data Ecosystem (Organisational)
Acknowledged Data Ecosystem (Distributed)
Collaborative Data Ecosystem (Federation)
Virtual Data Ecosystem (Coalition)
Reciprocal
Pooled
Type of Interdependence

**Fig. 2.9** Topology of data ecosystems [1]. Adapted from [59, 28]

- *Control of Key Data Resources*: Who controls the essential data resources in the data ecosystem? Does a single "Keystone" [57] actor control the key data resources that all others depend on? Alternatively, is control of the critical data resources spread across multiple actors in the data ecosystem?
- *Participant Interdependence*: The degree to which different participants in the data ecosystem must interact and exchange data for performing their activities. Reciprocal interdependence requires high levels of coordination among the participants, while pooled interdependence enables loose coupling among participants.

Drawing inspiration from the SoS classification by Maier [28] (including Virtual, Collaborative, Acknowledged, and Directed) and the business ecosystem topology by Koenig, we can (see Fig. 2.9) consider the different types of data ecosystems [1] that may exist within a smart environment and the nature of the relationships among the participants.

- *Directed Data Ecosystems*: Centrally controlled to fulfil a specific purpose. Typically found within an organisation setting or following a keystone model. Participants within a directed data ecosystem maintain an ability to operate independently, but their normal operational mode is subordinated to the centrally managed purpose of the data ecosystem.
- *Acknowledged Data Ecosystems*: Have defined objectives and pooled dedicated resources. The constituent systems retain their independent ownership and objectives. Changes in the data ecosystem are based on collaboration among the distributed participants.
- *Collaborative Data Ecosystems*: Participants interact voluntarily to fulfil an agreed-upon central purpose. The primary players collectively decide the means of enforcing and maintaining standards among the federations of participants.

- *Virtual Data Ecosystems*: Have no central management authority and no centrally agreed-upon purpose. Bottom-up coalitions of participants emerge from a virtual data ecosystem to pool decentralised resources to achieve specific goals.

Within a well-functioning data ecosystem, the participants are efficiently and effectively sharing across knowledge, value, and ecosystem boundaries. The nature of the ecosystem and the systems and their dynamics will affect the design and operation of the data ecosystem. To enable an intelligent systems data ecosystem, it is clear we will need to rethink some of the fundamentals of current intelligent system design approaches regarding governance, economics, and technical approaches.

### 2.7.4 Data Ecosystem Design

Several ecosystem design characteristics are detailed in Table 2.4. It is worth considering that multiple data ecosystems could exist at one time, and the operation of a data ecosystem can change depending on the circumstances. Concerning the design of intelligent systems, these design characteristics can affect the style of infrastructure that is needed to support data sharing within the data ecosystem, from data provided by a single dominant actor on their proprietary infrastructure, to a community, pooling their data in a managed open source data platform.

**Table 2.4** Data ecosystem design space

| Design characteristics | | Solution design space | |
|---|---|---|---|
| Governance | Control | Centralised | Decentralised |
| | Interdependence | Reciprocal | Pooled |
| | Structure | Authoritarian | Democratic |
| | Regulation | None | Enforceable |
| | Independence | Controlled | Autonomous |
| | Environment | Stable | Dynamic |
| Economic | Model | Pay | Free/sharing |
| | Connectivity | Keystone | Value network |
| | Data market | Single-sided | Multi-sided |
| | Collaboration | Competition | Cooperation |
| Technical | Infrastructure | Proprietary | Open |
| | Data availability | Closed | Open |
| | Privacy | Monitoring | Privacy-protecting |
| | Data formats | Homogeneous | Heterogeneous |
| | Data services | Exact | Approximate |

## 2.8 Iterative Boundary Crossing Process: Pay-As-You-Go

The *Iterative Boundary Crossing Processes* follow a socio-technical approach to accommodate iterations at crossing system boundaries. A key capability is the need to support a flexible, iterative approach that facilitates incremental agreements and investments among stakeholders. Pay-as-you-go data management approaches (such as dataspaces) are needed for technical concerns, while data ecosystem supports are needed to facilitate incremental transformations of political and organisational concerns.

### *2.8.1 Dataspace Incremental Data Management*

A dataspace is an emerging approach to data management that is distinct from current approaches. The dataspace approach recognises that in large-scale integration scenarios, involving thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources [2]. Within the dataspace paradigm, data management pushes the boundaries of traditional databases in two main dimensions [2]: (1) Administrative Proximity, which describes how data sources within a space of interest are close or far in terms of control; and (2) Semantic Integration, which refers to the degree to which the data schemas within the data management system are matched up. Dataspaces shift the emphasis to providing support for the co-existence of heterogeneous data that does not require a significant upfront investment into a unifying schema. Data is integrated on an "as-needed" basis with the labour-intensive aspects of data integration postponed until they are required. Dataspaces reduce the initial effort required to set up data integration by relying on automatic matching and mapping generation techniques. This results in a loosely integrated set of data sources. When tighter semantic integration is required, it can be achieved in an incremental "pay-as-you-go" fashion by detailed mappings among the required data sources. Dataspaces are described in further detail in Chap. 3. We have created the Real-time Linked Dataspace (RLD) (see Chap. 4) as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data and real-time stream and event processing capabilities to support a large-scale distributed heterogeneous collection of streams, events, and data sources [4].

The KVE Framework has outlined a high-level approach to support the exchange of knowledge among intelligent systems within a data ecosystem. In order to realise the sharing of knowledge between interconnected intelligent systems, there is a need for a data platform.

## 2.9 Data Platforms for Intelligent Systems Within IoT-Based Smart Environment

Platform approaches have proved successful in many areas of technology [60], from supporting transactions among buyers and sellers in marketplaces (e.g. Amazon), innovation platforms which provide a foundation on which to develop complementary products or services (e.g. Windows), to integrated platforms which are a combined transaction and innovation platform (e.g. Android and the Play Store).

The idea of large-scale "data" platforms have been touted as a possible next step for the development of smart environments [1] and data ecosystems. An ecosystem data platform would have to support continuous, coordinated data flows, seamlessly moving data among intelligent systems. The design of infrastructure to support data sharing and reuse is still an active area of research. In order to understand the general requirements necessary to share data, we examine the "FAIR Data" principles [61] that have been defined to support data reuse within the scientific community. Then, to understand the specific data sharing requirements for an intelligent systems data ecosystem, we examine the data management needs of five different IoT-based smart environments.

### *2.9.1 FAIR Data Principles*

In order to improve the data infrastructure supporting the reuse of research data, a group of stakeholders from academia, industry, funding agencies, and research publishers proposed a set of principles known as the FAIR Data Principles [61]. The FAIR principles are Findability, Accessibility, Interoperability, and Reusability with a detailed breakdown of the principles provided in Table 2.5. The objective of the principles is to act as a set of guidelines to data producers and publishers to maximise the reusability of research data. The FAIR principles are designed to enable proper data management to support knowledge discovery and innovation, and the subsequent data and knowledge integration and reuse. The principles define the goals of good data management and stewardship practices to improve its reusability. The principles can influence the design of algorithms, tools, and workflows for research data. The broad application of the principles can lead to a data research ecosystem that supports extracting maximum benefit from research investments by ensuring transparency, reproducibility, and reusability. Within the context of this work, we use the principles as a high-level guide for the design of a data platform to support knowledge sharing between intelligent systems within a smart environment.

**Table 2.5** FAIR guiding principles for scientific data management and stewardship [61]

| |
|---|
| To be findable |
| F1. (Meta)data is assigned a globally unique and persistent identifier |
| F2. Data is described with rich metadata (defined by R1 below) |
| F3. Metadata clearly and explicitly includes the identifier of the data it describes |
| F4. (Meta)data is registered or indexed in a searchable resource |
| To be accessible |
| A1. (Meta)data is retrievable by their identifier using a standardised communications protocol |
| A1.1 The protocol is open, free, and universally implementable |
| A1.2 The protocol allows for an authentication and authorisation procedure, where necessary |
| A2. Metadata is accessible, even when the data is no longer available |
| To be interoperable |
| I1. (Meta)data uses a formal, accessible, shared, and broadly applicable language for knowledge representation |
| I2. (Meta)data uses vocabularies that follow FAIR principles |
| I3. (Meta)data includes qualified references to other (meta)data |
| To be reusable |
| R1. (Meta)data is richly described with a plurality of accurate and relevant attributes |
| R1.1. (Meta)data is released with a clear and accessible data usage license |
| R1.2. (Meta)data is associated with detailed provenance |
| R1.3. (Meta)data meets domain-relevant community standards |

## 2.9.2 Requirements Analysis

Over the past years, we have been involved in a number of projects [4, 18, 62, 63] concerned with next-generation data platforms for intelligent systems within smart environments. The smart environments focused on intelligent energy and water management with varying sizes of data ecosystems. The five pilots are:

- *Smart Airport*: Linate airport in Milan represents large-scale commercial energy and water consumer for use from washing activities, toilets, restaurants, and irrigation, flight operations, to safety-critical infrastructure for emergency response. Linate targets a variety of users, from the company's employees (including executives, operational managers, and technical staff), to passengers. The variety of sensors used in the airport requires the management of heterogeneous events and their availability to applications in near-real-time. Significant contextual data from the airport's operational legacy systems is needed to process the events for decision-making.
- *Smart Office*: The Insight Building was built in the 1990s without a building management system and has been retrofitted with energy sensors. As typically in an organisation, Insight has several information systems that run its operations, including finance and enterprise resource planning, budgeting, and office IT assets. These enterprise systems can help in identifying energy wastage and promoting conservation actions within the office.

- *Smart Homes*: The Municipality of Thermi in Greece provides a residential smart water pilot with a representative sample of ten domestic residences. The target users are the residents (both adults and children), municipality management, a developer community for smart home "Apps," research scientists, and the local water utility. Data from IoT devices in each home needs to be managed in a near-*real-time* manner to provide feedback to users on their water consumption. Secure sharing of data with both the research and developer community is needed.
- *Mixed Use*: The Engineering Building at NUI Galway in Ireland is a state-of-the-art smart building with significant numbers of sensors and actuators. Target users include academic staff, managers, technicians, researchers, and students. This smart environment is designed to be a "living laboratory" where the building itself is an interactive teaching tool where students can utilise data from the environment in their projects and research works. Making data easily reusable by occupants in the environment is an essential requirement.
- *Smart School*: Coláiste na Coiribe is a newly constructed Irish language secondary school. The school accommodates students aged between 12 and 18, together with teaching and operational staff. The school has been fitted with a commercial state-of-the-art building management system to manage its energy and water consumption. A key challenge is to customise the communication of energy and water data for the diverse range of school stakeholders.

For each of these five smart environments, a system analysis was performed to identify the functional and non-functional information processing and sharing requirements. These requirements complement the FAIR principles by including concrete requirements for data processing, querying, and data ecosystem support, including the need for iterative, incremental processes. The following common data platform requirements were identified across the pilots [4]:

- *Pay-As-You-Go Data Integration, Accessibility, and Sharing*: Each smart environment contains potentially thousands of data sources from sensors and things to legacy information systems. Harnessing this data is critical to enabling the smart environment. Challenges include the integration of multiple formats and semantics, discoverability and access, and data re-use and sharing in a low-cost and incremental manner [33, 64–67]. This high-level requirement can be broken down into a set of technical requirements:
  - *Standard data syntax, semantics, and linkage:* Facilitate integration and sharing, ideally with open standards and non-proprietary approaches.
  - *Single-point data discoverability and accessibility:* Allow the organisation and access to datasets and metadata through a single location.
  - *Incremental data management:* Enable a low barrier to entry and a pay-as-you-go paradigm to minimise costs.
- *Secure Access Control*: Support data access rights to preserve the security of data and privacy of users in the smart environment. Access control is needed at both the level of the data source and at the level of the data item (i.e. entity-value).

- *Real-time Data Processing and Historical Querying*: Each environment requires support for the real-time processing of data generated from sensors and things within the environments. This requirement can be broken down into two technical requirements:
  – *Real-time data processing:* Including ingestion, aggregation, and pattern detection within event streams originating from sensors and things in the smart environment.
  – *Unified querying of real-time data and historical data:* Provide applications and end-users with a holistic queryable state of the smart environment at a latency suitable for user interaction.
- *Entity-Centric Data Views*: Intelligent applications and end-users need to be able to explore and query the data from an entity perspective, such as energy or water usage in a specific building zone. The raw data generated by things (e.g. a smart tap) within the environments often only report on the observed values of a property (e.g. water consumption). Thus, the raw sensor/thing data may require additional contextual information, such as the location of the sensor [64–66]. This high-level requirement can be broken down into two technical requirements:
  – *Entity management:* The storage, linkage, curation, and retrieval of entity data, such as users, zones, and locations
  – *Event enrichment:* Enhancement of sensor/things streams with contextual data (e.g. entities) to make the stream data more encapsulated and useful in downstream processing

The level of importance of these common data requirements varies within each pilot as detailed in Table 2.6. Many other requirements were identified within the smart environments, including interoperability of devices and network protocols, user profiling, the resilience of remote sensors, and advanced privacy-preserving analytics.

**Table 2.6** Level of importance of common data platform requirements [4]

| Requirements | Smart Airport | Smart Office | Smart Home | Mixed Use | Smart School |
|---|---|---|---|---|---|
| Standard data syntax, semantics, and linkage | High | Medium | Low | Medium | Medium |
| Single-point data discoverability and accessibility | High | Medium | High | High | Medium |
| Incremental data management | High | High | Low | High | Medium |
| Secure access control | High | High | High | High | Medium |
| Real-time data processing | High | High | Medium | High | High |
| Unified querying of real-time data and historical data | High | High | High | High | High |
| Entity management | High | High | Medium | High | Medium |
| Event enrichment | High | High | High | High | Medium |

## 2.10 Summary

The digital transformation is creating a data ecosystem with data on every aspect of our world spread across a range of information systems. Data ecosystems present new challenges to the design of intelligent systems and System of Systems that demands a reconsideration of how we deal with the needs of large-scale, data-rich smart environments. In this chapter, we have explored the barriers to the sharing of knowledge among intelligent systems within a smart environment and how they can be overcome with the capabilities within the Knowledge Value Ecosystem (KVE) Framework. The implementation of these capabilities was explored using linked data, knowledge graphs, and data value chains, which provide solid foundations for tackling system boundaries of knowledge exchange among systems. Finally, the chapter examined the need for data platforms to support the sharing of data between intelligent systems within a data ecosystem and identified common data platform requirements.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

# Chapter 3
# Dataspaces: Fundamentals, Principles, and Techniques

**Keywords** Dataspaces · Best-effort information · Approximation · Incremental data management

## 3.1 Introduction

A dataspace is an emerging approach to data management which recognises that in large-scale integration scenarios, involving thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources. Data is integrated on an "as-needed" basis with the labour-intensive aspects of data integration postponed until they are required. Dataspaces reduce the initial effort required to set up data integration by relying on automatic matching and mapping generation techniques. This results in a loosely integrated set of data sources. When tighter semantic integration is required, it can be achieved in an incremental "pay-as-you-go" fashion by detailed mappings between the required data sources. This chapter introduces dataspaces and the fundamentals of "best-effort" data management.

The chapter is structured as follows: Sect. 3.2 discusses big data and the challenge of data integration with the long tail of data variety, Sect. 3.3 examines the cost of data management, and Sect. 3.4 addresses the emerging trend of approximate, best-effort and "Good Enough" approaches to data management. Section 3.5 provides a detailed explanation of the fundamentals of dataspaces, including their principles and a comparison to contemporary data management approaches. Section 3.6 covers dataspace support platforms, support services, life cycle, and specific implementations. Section 3.7 details the technical challenges for dataspaces, Sect. 3.8 sets out ongoing research challenges for dataspaces, and finally, a summary is provided in Sect. 3.9.

© The Author(s) 2020
E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_3

## 3.2 Big Data and the Long Tail of Data

The emergence of new platforms for decentralised data creation such as the Internet of Things and edge computing, the increasing availability of open data on the web [68], and the increasing number of data sources inside organisations [69] bring an unprecedented volume of data to be managed. In addition to coping with the volume of data, data consumers in the big data era need to cope with data variety where data is created under different contexts and requirements [25]. Consuming data comes with the intrinsic cost of repurposing, adapting, and ensuring data quality for its new context. Data at large scales coming from distributed sources can be erroneous, inconsistent, and incomplete for some users' requirements. Jagadish et al. state that "Big Data increasingly includes information provided by increasingly diverse sources, of varying reliability. Uncertainty, errors, and missing values are endemic, and must be managed" [70].

The growth in the number of data sources and the increasing scope of information systems leads to a *long tail of data variety* [71]. The long tail of data variety (see Fig. 3.1) reflects the distribution of the frequency of use of conceptual elements: in a large domain of interest few entities and attributes have a high frequency of use, followed by a long tail distribution of entities and attributes which have lower frequencies of use. While some concepts are central across many different areas, most of the concepts are specific to a context. In the scientific domain, for example, the long tail of scientific data [72] reflects the conceptual distribution of scientific data.

Traditional relational data management environments were focused on data that mapped to popular business processes and were regular enough to fit into a relational model. The long tail of data variety expresses the shift towards the expanding coverage of data that must be managed; it is less frequently used, more decentralised, and less

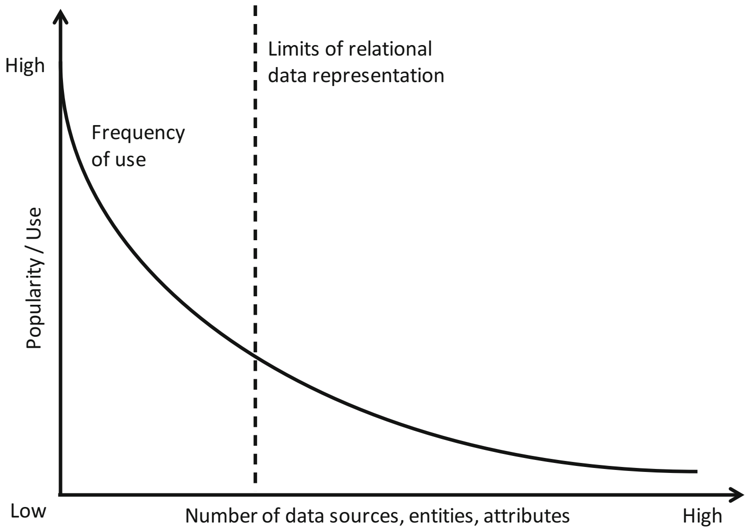


**Fig. 3.1** The long tail of data variety. Adapted from [71]

structured. Given this shift in the data landscape, there has been an evolution in the information processing landscape to meet these new challenges. Big Data technologies are moving towards supporting the long tail of data variety to enable consumers to have a richer and more comprehensive model of their domain that they can *search*, *query*, *analyse,* and *navigate*. However, managing this long tail of data comes with a cost.

## 3.3 The Changing Cost of Data Management

Historically, the construction of information systems and databases has evolved following a model dependent on the cost of formalising a domain and the associated value derived from the efficiency gain. Data management practices for organisations have prioritised the formalisation (conceptualisation) of domains within a centralised model with high levels of control, which have resulted in high data management costs. Propelled by the growth of data and the increasing number of systems and devices producing data, data management requirements are shifting towards the need to cope with data which is generated in a decentralised manner [73, 74], data which is intrinsically heterogeneous, and data with varying levels of structure and different contexts. These trends are contributing to the long tail distribution of data variety.

According to Brodie and Liu [69]:

> The consistency of all views of the same tuple leads the underlying belief in a single version of the truth and the concept of a global schema. The dramatic success of relational technology has propelled data modelling and management requirements beyond the modelling and processing capabilities of the relational technology. The phrase ‘single version of the truth’ seems intuitively correct and may assure in a confusing world, but it is almost entirely false in the real world. The underlying assumption of the relational world is not just semantic homogeneity but also ontological homogeneity while in reality, semantic heterogeneity dominates. Data management vendors promote the ‘single version of truth’ assumption as a highly desirable objective and something that their products can provide. Our Digital Universe is no longer a semantically homogeneous set of a few databases but Information Ecosystems of 100s or 1000s of semantically heterogeneous databases to be managed and integrated collectively [69].

Franklin et al. [2] highlight that “in data management scenarios today it is rarely the case that all the data can be fit nicely into a conventional relational [database]”. Diving deeper into the practical challenges of managing decentralised and heterogeneous data, Franklin et al. [2] examine the problem along two problem dimensions (see Fig. 3.2):

- *Administrative Proximity:* Describes how data sources within a space of interest are close or far in terms of control. A close control means that many assumptions can hold concerning guarantees such as data quality and consistency, while a far control refers to a loosely coupled environment and a lack of coordination on the data sources within the data management system.
- *Semantic Integration:* Refers to the degree of how much the data schemas within the data management system are matched up. That includes, for example, the types, attributes, and names used within the data sources. On one end of the

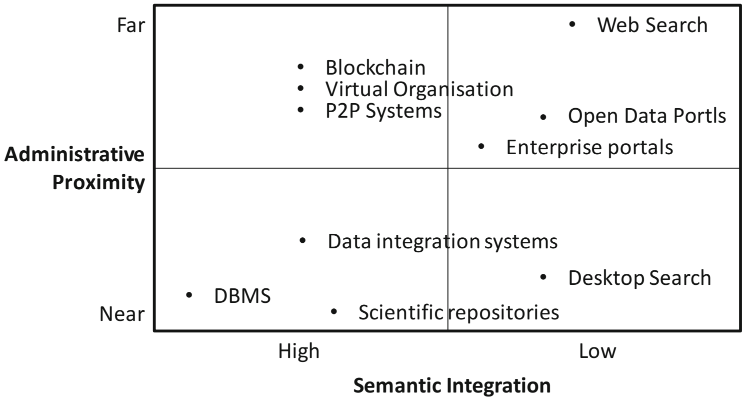


**Fig. 3.2** Data management approaches. Adapted from [2]

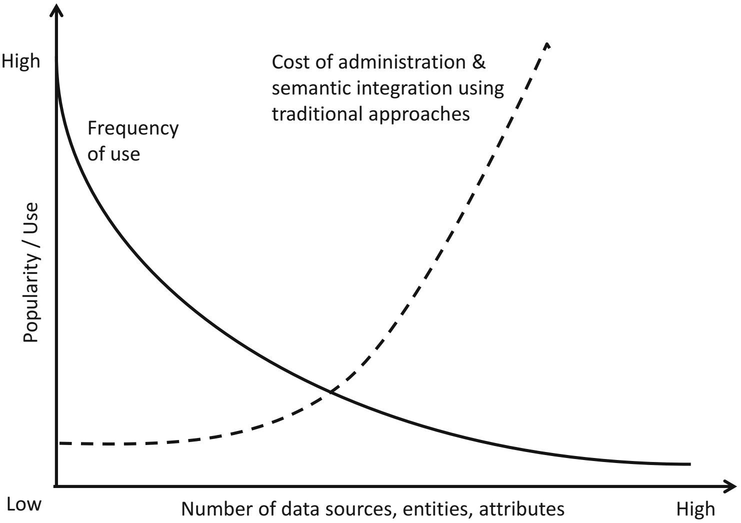


**Fig. 3.3** Data management cost associated (administration and semantic) with increasing numbers of schema and sources. Adapted from [71]

spectrum, all data conform to an agreed-upon schema, while on the other end of the spectrum schema information is lacking. This dimension is relevant to how much semantically rich querying can be done.

In this view, data management should be considered as a task that takes place in a spectrum defined by these two dimensions. Within a Database Management System (DBMS), administrative control and semantic homogeneity are key to the data modelling, querying, and integration approaches that depend on the relational data model. This results in a high upfront cost for DBMS, which only gets more significant when dealing with the long tail of data, as illustrated in Fig. 3.3.

The need to develop more effective and efficient approaches for dealing with decentralised semantically heterogeneous environments is highlighted by the Lowell Self-Assessment report [75], which is a roadmap for the future of research in the database community: "A semantic heterogeneity solution capable of deployment at Web scale remains elusive. At Web scale, this is infeasible, and query execution must move to a probabilistic world of evidence accumulation and away from exact answers. Therefore, one must perform information integration on-the-fly over perhaps millions of information sources" [75].

## 3.4 Approximate, Best-Effort, and "Good Enough" Information

Not every situation or decision requires information that is 100% fresh and accurate [76], as many information consumers can efficiently work with information at a lower threshold. A perfect result is not always necessary, while a lower-quality or less-than-optimal result is sufficient [77]. By relaxing the need for computing to the highest quality, approximate approaches can be used to improve the throughput and response time of services.

Many applications that require data processing can tolerate a reduced quality in the result of the data analysis. Consider the rise of "Recognition" applications, a new class of popular mobile edge applications that range from recognising a single user's surroundings and augmenting it with information, advice, and decision support, to analysing an array of images from traffic cameras within a smart city to manage traffic. Recognition applications may have the possibility to trade off the accuracy of analysis with the responsiveness and computation necessary for the analysis.

The Pareto Principle (or the 80/20 rule) has wide application in many areas from economics and market analysis to business strategy, where it has been observed that 20% of the effort delivers 80% of the results. Within computer science, the principle has been observed within many problems from fixing bugs to writing code. The principle can help us to prioritise actions, for example, focus on the 20% of software bugs that cause 80% of the system crashes. Data management approaches can take advantage of the fact that many users and applications can tolerate approximate results; a trade-off between exact and approximate results can minimise data integration costs and the response time (both network and compute), and maximise throughput. Relaxing the need for maximum quality reduces the required data integration and computation workload, enables a significant reduction of response time, and increases throughput.

Approximate approaches can reduce semantic integration costs due to their ability to deal with the uncertainties of semantics. Approximate approaches can operate in environments with low-cost agreements on administration proximity and semantic integration, and at the same time, achieve acceptable levels of precision and recall comparable to exact models. These characteristics make these approaches suitable for tackling the long tail of data variety.

Approximate “good enough” approaches are distinguished by a matching model that is not Boolean and supports one form or another of uncertainty, probability, or ranking of the results. This gives the matching model more flexibility to deal with data heterogeneity and thus, improves its ability to address lower levels of administrative control and semantic integration.

The classic relational paradigm defined the use of structured query languages (such as SQL) as the primary mechanism for user–data interaction. SQL offers crisp and accurate answers for relatively small numbers of homogeneous sources [76], typically managed in a centralised and coordinated manner. Within the approximate model, the underlying assumptions of user-data interaction and the use of structured queries to cope with the long tail of data were revisited by [39, 76]:

**From:** *Clean, semantically homogenous and centralised schema*

**To:** *Semantically heterogenous and decentralised schema*

As schemas are managed in a decentralised way, different conceptualisations may exist in the same schema. “We can no longer pretend to live in a clean world . . . .” [76]. “Unless the reader of a message or document is specifically programmed for it, there will likely be confusion. The meaning of the message, the interpretation of its fields, and much more, will be subject to approximation and a loss of clarity. Different companies, different countries, and even different regions within a country have different understandings of data” [76].

**From:** *Manual query-schema mapping*

**To:** *Automatic query-schema mapping*

Most of the interaction with structured data is dependent on manual mapping among elements of a structured query and schema elements. With the growth in the schema-size and the number of available data sources, the cost associated with this manual mapping process becomes prohibitive (see Fig. 3.3) requiring automated and semi-automated query-mapping techniques.

**From:** *Absolute precision/full recall in a single query*

**To:** *Relaxed precision/recall in multiple queries*

As schemas grow and as users cross database boundaries, the cost associated with building structured queries grows exponentially. In this scenario, the expectation of getting a correct and complete answer in a single interaction should be exchanged by approximate answers which are obtained through multiple interactions. As Helland states [76] “Too much, too fast—you need to approximate.”

By creating data management approaches that utilise approximate and best-effort techniques, it is possible to reduce the cost of dealing with the long tail of data variety. Approximate approaches trade off administrative and integration costs with reduced accuracy of results. “Best-Effort” thinking is at the core of the new dataspace paradigm of data management (Fig. 3.4).

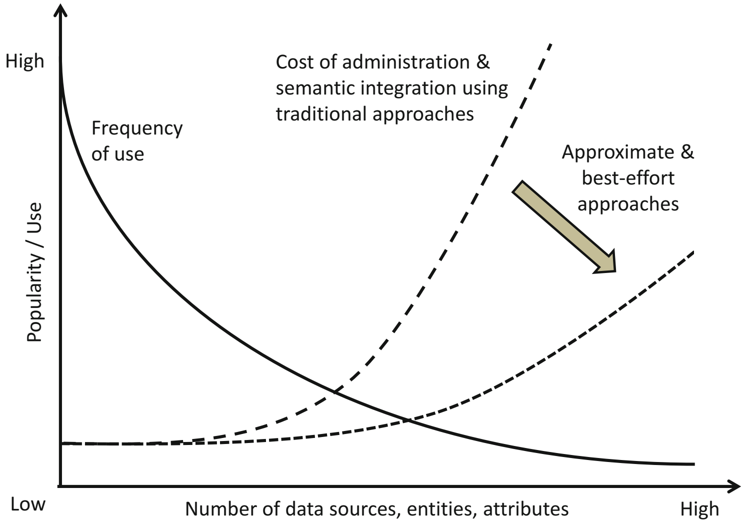


**Fig. 3.4** The long tail of data and the improved scalability of data management with approximate and best-effort approaches

## 3.5 Fundamentals of Dataspaces

A dataspace is an emerging approach to data management that is distinct from current approaches. The dataspace approach recognises that in large-scale integration scenarios, involving thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources [2]. Dataspaces are not a data integration approach [2]; they shift the emphasis to providing support for the co-existence of heterogeneous data that does not require a significant upfront investment into a unifying schema. Data is integrated on an "as-needed" basis with the labour-intensive aspects of data integration postponed until they are required. Dataspaces reduce the initial effort required to set up data integration by relying on automatic matching and mapping generation techniques. This results in a loosely integrated set of data sources. When tighter semantic integration is required, it can be achieved in an incremental "pay-as-you-go" fashion by detailed mappings among the required data sources. This section details the fundamentals of the dataspace paradigm, including their core principles, comparison to existing approaches, support platform, data services, life cycle, and research challenges.

### *3.5.1 Definition and Principles*

First introduced by Franklin, Halvey, and Maier in 2005 [2], a dataspace can contain all the data sources for an organisation regardless of its format, location, or model.

**Table 3.1** Definitions of a "Dataspace" from literature

| Definition | Source |
|---|---|
| "Dataspaces are not a data integration approach; rather, they are more of a data co-existence approach. The goal of dataspace support is to provide base functionality over all data sources, regardless of how integrated they are." | [78] |
| "A dataspace system manages the large-scale heterogeneous collection of data distributed over various data sources in different formats. It addresses the structured, semi-structured, and unstructured data in coordinated manner without presuming the semantic integration among them." | [79] |
| "to provide various of the benefits of classical data integration, but with reduced upfront costs, combined with opportunities for incremental refinement, enabling a "pay-as-you-go" approach." | [80] |
| "enable agile data integration with much lower upfront and maintenance costs." | [81] |
| "A dataspace system processes data, with various formats, accessible through many systems with different interfaces, such as relational, sequential, XML, RDF, etc. Unlike data integration over DBMS, a dataspace system does not have full control on its data, and gradually integrates data as necessary." | [82] |
| "Dataspace Support Platforms envision data integration systems where the amount of upfront effort is much smaller. The system should be able to bootstrap itself and provide some useful services with no human intervention. Over time, through user feedback or as sources are added and the data management needs become clearer, the system evolves in a pay-as-you-go fashion." | [83] |
| "Dataspace is defined as a set of participants and a set of relationships among them." | [84] |

Each data source (e.g. database, CSV, web service) in the dataspace is known as a participant. The dataspace can model the relations (or associations) between data in different participants. In its purest form, a dataspace is a set of participants and the inter-relations between them [2]. The modelling of the dataspace can capture different types of relations among participants, from the mapping of the schemas between two participants to capturing that Participant A is a replica of Participant B.

The dataspace concept has gained traction with a number of different groups exploring its usefulness for managing data from different domains and investigating the design of support services. These works have provided a number of definitions for a dataspace as captured in Table 3.1; most of these build on the initial concept from [2].

Dataspaces, as a paradigm of data management, push the boundaries of traditional databases along the dimensions of administrative proximity and semantic integration, as discussed in Sect. 3.3. Within a dataspace, data sources are not tightly controlled, and full semantic integration is not guaranteed. Data management within a dataspace is defined by different principles, as described by Halevy et al. [78]:

- A dataspace must deal with data and applications in a wide variety of formats accessible through many systems with different interfaces. A dataspace is required to support all the data rather than leaving some out because they do not yet conform to a specific schema or data constraint.

**Table 3.2** DBMS vs. Dataspace. Adapted from [85]

| | DBMS | Dataspace |
|---|---|---|
| Model | Relational | All |
| Formats | Homogeneous | Heterogeneous |
| Schema | Schema first, data later | Data first, schema later or never |
| Control | Complete | Partial |
| Leadership | Top-down | Top-down/Bottom up |
| Query | Exact | Approximate |
| Integration | Upfront | Incremental |
| Architecture | Centralised | Decentralised |
| Real-time data processing | No | Applicable |

- A dataspace must provide an integrated means of search, query, update, and administration. The dataspace does not subsume participant data sources and is not in full control of the data. The same data may also be accessible and modifiable through an interface native to the system hosting the data.
- Queries to a dataspace may offer varying levels of service, and in some cases may return best-effort or approximate answers. For example, when individual data sources are unavailable, the best answers available are returned using the data accessible at the time of the query.
- The dataspace must provide pathways to improve the integration among the data sources in a “pay-as-you-go” fashion.

### *3.5.2 Comparison to Existing Approaches*

Data sources in a dataspace co-exist, and co-evolve over time, and are not subsumed by a rigid data management system. This is in stark contrast to the traditional data management approach based on relational databases. A comparison of the dataspace paradigm to traditional DBMS is provided in Table 3.2.

## 3.6 Dataspace Support Platform

The goal of a Dataspace Support Platform (DSP), as detailed in Fig. 3.5 [2], is to provide a set of common related support services to all data sources within the dataspace (e.g. keyword search). The DSP provides a base functionality needed for data integration that enables developers to focus on application-specific challenges instead of the common data integration tasks faced when working with multiple data sources. To achieve this goal, the DSP must support all the data in the dataspace requiring it to work, with a large variety of data formats and system interfaces. A dataspace does not host data; the data resides in their native systems. As such, a

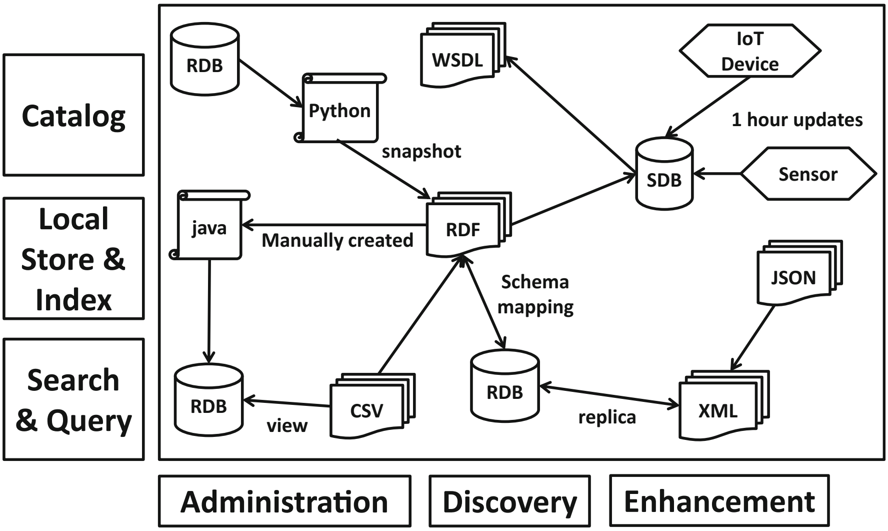


**Fig. 3.5** A dataspace support platform. Adapted from [2]

dataspace is not in full control of the data and may only provide weak guarantees of consistency and durability. When stronger guarantees are desired, more effort can be put into making agreements among the various systems. To this end, a DSP must provide tools to support the tighter integration of data in a pay-as-you-go manner. As a result of the varying levels of data integration, the DSP offers varying levels of service and often will only be able to provide best-effort or approximate results using the data accessible at the time of the query [2].

## *3.6.1 Support Services*

Services within a DSP need to support heterogeneous data types and multiple access methods to participants within the dataspace. A core set of support services, as identified by [2] are:

- *Catalog:* A catalog is an inventory of data elements from participants containing basic information about each one, including source, name, location, size, creation date, and owner. A catalog service can provide a basic browse interface across the dataspace for users. The catalog is a core infrastructure that is used by other dataspace support services.
- *Search:* Search is the primary mechanism used by end-users to deal with large collections of unknown data. Search is based on a similarity analysis of data that results in a ranked list of results relevant to an end-user's keywords. The search service should examine all the contents of a dataspace, including metadata. Search interfaces within a DSP should support the interactive refinement of the

results (e.g. facets) so that users can explore a dataset and incrementally improve the search results. The process of refinement could result in a database-style structured query.
- *Query:* The level of support for expressive queries will vary across participants in the dataspace, as some will provide expressive query languages, while others will only have limited interfaces for querying (e.g., web services). The query service of a DSP needs to provide for (1) metadata queries to support the discovery of data sources, (2) monitoring of data sources, (3) local storage and indexing to support associations among participants, increase query expressivity on sources with limited querying functionality, and improve data source availability.
- *Discovery:* It locates participants in the dataspace and supports the creation of relationships among them in an incremental manner.

Not every participant in a dataspace will support all DSP functions. Thus, there will be the need to extend data sources in various ways (e.g. search and query). This requires the services to support dataspace participants in an incremental manner that can be applied in real time to existing as well as new participants joining the dataspace. The incremental nature of the support services is a core enabler of the pay-as-you-go paradigm in dataspaces.

### 3.6.2 Life Cycle

Similar to any other data management approach, a dataspace has a life cycle of operation. Hedeler et al. [80] have proposed a conceptual life cycle for dataspaces, illustrated in Fig. 3.6, consisting of seven phases with transitions between phases. As dataspaces can be used within different contexts, only a subset of the phases and transitions in the life cycle may be relevant to a specific implementation or deployment. The key phases in the life cycle are:

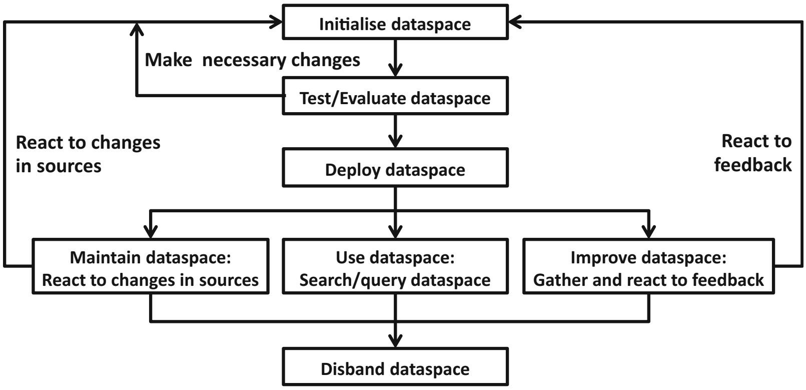

Initialise dataspace
Make necessary changes
Test/Evaluate dataspace
React to changes in sources
React to feedback
Deploy dataspace
Maintain dataspace: React to changes in sources
Use dataspace: Search/query dataspace
Improve dataspace: Gather and react to feedback
Disband dataspace

**Fig. 3.6** Conceptual life cycle of a dataspace [80]

- *Initialise:* Identification of the data sources to be accessible within the dataspace and the integration of those resources. Sub-phases can include identifying data sources, designing integration schema, identifying matchings, deriving mappings, and deriving integration schemas.
- *Test/Evaluate:* Testing and evaluation of the dataspace before deployment.
- *Deployment:* Enabling access to the dataspace for users and applications, including the deployment of the dataspace on its physical computing infrastructure.
- *Use:* Support users and applications accessing and using the dataspace, including the search and query of sources in the dataspace.
- *Maintain:* Supporting changes to the participants in a dataspace, including adding and removing a source.
- *Improve:* Improving the dataspace during its operation, including the integration of participants. Explicit and implicit user feedback can be a key source of improvement for the dataspace.
- *Disband:* Gracefully close the dataspace by removing the participants and free resources.

As noted by [80], the phases Use, Maintain, and Improve are co-existing to support the “pay-as-you-go” model of data management.

### *3.6.3 Implementations*

The dataspace approach has been implemented using several different technology stacks and used within a number of different contexts, including:

- *Personal Dataspace:* Focusing on the management of the information on a person across various sources, from their desktop [86] and mobile devices to their presence on social networks. Works include: iDM [87], SEMEX [88, 89], iMeMeX [90, 91], CoreSpace [92], and PDSP [93].
- *Scientific Dataspace:* Working with distributed sources of scientific data including astronomy data [94], biomedical data [81, 95], life sciences data with ALADIN [96] and LinkedScales [97], and process materials with the Virtual Data Space [98, 99].
- *Enterprise/Industrial Dataspace:* Targets the use of a dataspace to bring together data from different organisations within the context of energy management [100], air travel (Airbus Skywise), Industry 4.0 [101], or a digital library [102].
- *Global/Web Dataspaces (Web of Data):* Efforts at global or web-scale dataspaces include PayGo [103] and OCTOPUS [104]. The use of linked data technologies to publish data on the web is enabling the linkage of records in distinct databases that can be viewed as a global dataspace [43]. They can have a multi-domain nature such as the Linked Open Data Cloud [105], or a domain-specific purpose, such as financial data [56].
- *Software Development:* The use of a dataspace to support software artefact management [106].

- *Internet of Things/Smart Environment Dataspaces:* Recent works have investigated the use of dataspaces to support the Internet of Things or smart environments [4, 107] and the data they produce.

## 3.7 Dataspace Technical Challenges

The key technical challenges to realising a dataspace as identified by Halevy et al. [78] centre around the need to answer queries, to introspect on the content of the answers, and to leverage human interaction to enhance the semantic relationships within a dataspace. In this section, we briefly summarise these challenges as detailed by Halevy et al. [78].

### *3.7.1 Query Answering*

In order to understand the fundamental challenges of querying a dataspace, we briefly summarise modes in which we expect users to interact with a dataspace.

**Participants and Relationships** Dataspaces are modelled as a rich collection of participants and relationships that contain all of the information relevant to a particular organisation or entity regardless of its format, location, or data repository.

**Queries** Since multiple data models need to be supported within a dataspace, queries will also come in a variety of languages; from keyword searchers to structured forms and formal query languages. The dataspace support platform needs to provide mechanisms for executing queries in different languages and return results from all the relevant sources in the dataspace.

**Answers** Answers to queries within a dataspace follow a best-effort model which is different from queries over traditional databases. Answers can come in the following forms [78]:

- *Ranked:* A ranked set of answers to structured queries and/or keyword search. Rankings may be based on different methods (e.g. relatedness, similarity) or approximate matchings from different sources.
- *Heterogeneous:* Answers can come from different sources using different data models and schemas.
- *Sources as Answers:* Answers can include pointers to sources where additional answers can be found.
- *Iterative:* User query interaction follows an iterative approach with the user posing a sequence of queries, each being a refinement or modification of the previous query.
- *Reflection:* Completeness of the query coverage and its accuracy is included in the answer.

- In situ*:* Answers can be references to, rather than copies of, the data.

**Query Answering model**

In order to support the above query answering model, dataspaces need to overcome a number of challenges [78].

*Challenge 1: Develop a formal model for studying query answering in dataspaces*

- *C1.1:* Develop intuitive semantics for answering a query that takes into consideration a sequence of earlier queries leading up to it.
- *C1.2:* Develop a formal model of information gathering tasks that include a sequence of lower-level operations on a dataspace.
- *C1.3:* Develop algorithms that given a keyword query and a large collection of data sources, will rank the data sources according to how likely they are to contain the answer.
- *C1.4:* Develop methods for ranking answers that are obtained from multiple heterogeneous sources (even when semantic mappings are not available).

**Obtaining Answers**

Within a dataspace, the answers to queries come from heterogeneous data that may use different terms at both the schema level and the data level. Dataspaces do not rely on semantic mappings, and even when mappings exist, they may be partial or approximate. This poses a significant challenge to answering queries in a dataspace [78].

*Challenge 2: Develop methods for answering queries from multiple sources that do not rely solely on applying a set of correct semantic mappings*

- *C2.1:* Develop techniques for answering queries based on the following ideas, or combinations thereof:
  - Apply several approximate or uncertain mappings and compare the answers obtained by each.
  - Apply keyword search techniques to obtain some data or some constants that can be used in instantiating mappings.
  - Examine previous queries and answers obtained from data sources in the dataspace and try to infer mappings between the data sources. Whenever we have access to queries that span multiple data sources, try to infer from them how the sources are related (e.g. the join attributes should provide some hint of common domains).
- *C2.2:* Develop a formal model for approximate semantic mappings and for measuring the accuracy of answers obtained with them.
- *C2.3:* Given two data sets that use the same terminology but for different data models, develop automatic best-effort methods for translating a query over one data set onto the other.

## 3.7.2 Introspection

The data in a dataspace will be uncertain and often inconsistent. The uncertainty will increase due to the best-effort query answering. Answers can be different, depending on the level of latency and completeness within the dataspace. It is often the case that inconsistencies lead to a very particular kind of uncertainty: which of a set of conflicting data values is correct. Both uncertainty and inconsistency need to be resolved, and lineage is often the only method available [78]. A dataspace needs to be able to introspect about lineage, uncertainty, and inconsistency.

**Lineage, Uncertainty, and Inconsistency**

The challenge is for a dataspace to provide a single unified mechanism for modelling uncertainty, inconsistency, and lineage [78].

*Challenge 3: Develop formalisms that enable modelling uncertainty, inconsistency, and lineage in a unified fashion.*

- *C3.1:* Develop formalisms that capture uncertainty about common forms of inconsistency in dataspaces.
- *C3.2:* Develop formalisms for representing and reasoning about external lineage.
- *C3.3:* Develop a general technique to extend any uncertainty formalism with lineage and study the representational and computational advantages of doing so.
- *C3.4:* Develop formalisms where uncertainty can be attached to tuples in views and view uncertainty can be used to derive uncertainty of other view tuples.

**Finding the Right Answers**

Given the challenges of lineage, uncertainty, and inconsistency of query answers in the dataspace, it becomes necessary to determine a "good" answer. Candidate answers can differ along multiple dimensions [78], including:

- Relevance to the query
- Certainty of the answer (or whether it contradicts another answer)
- Completeness and precision requested by the user
- Maximum latency required in answering the query

*Challenge 4: Define metrics for comparing the quality of answers and answer sets over dataspaces, and efficient query processing techniques.*

- *C4.1:* Develop query-language extensions and their corresponding semantics that enable specifying preferences on answer sets along the dimensions of completeness and precision, certainty and inconsistency, lineage preferences and latency.
- *C4.2:* Define notions of query containment that take into consideration completeness and precision, uncertainty and inconsistency and lineage of answers, and efficient algorithms for computing containment.
- *C4.3:* Develop methods for efficient processing of queries over uncertain and inconsistent data that conserve the external and internal lineage of the answers. Study whether existing query processors can be leveraged for this goal.

### 3.7.3 Reusing Human Attention

Every participant within a dataspace is provided with a basic level of service when they join it. Over time the dataspace should evolve by forming tighter semantic integration between participants as needed. One key mechanism to achieve this goal is to leverage users' attention as they interact with a dataspace. By analysing the interaction of a user with different participants within the dataspace, we can gain knowledge about the relationships between data sources [78].

*Challenge 5: Develop methods that analyse users' activities when interacting with a dataspace and create additional meaningful relationships between sources in a dataspace or other enhancements to the dataspace.*

- *C5.1:* Develop techniques that examine collections of queries over data sources and their results to build new mappings between disparate data sources.
- *C5.2:* Develop algorithms for grouping actions on a dataspace into tasks.
- *C5.3:* Develop facilities for explicit enhancement of dataspace information that give a high return on the investment of human attention.
- *C5.4:* Develop a formal framework for learning from human attention in dataspaces.

## 3.8 Dataspace Research Challenges

The dataspace approach to data management raises several research challenges that need to be tackled to create effective and efficient DSPs. Research challenges include:

- *Data Models, Search, and Query:* A dataspace needs to support the various data models and the different query languages of the participants with varying levels of query expressivity [2]. Research is needed to support a broad view of data modelling [83, 97, 108, 109] and to enable querying over the heterogeneous data models, from basic queries to context-based queries [110] and schema-agnostic question answering systems [111]. There has been limited work on addressing the requirements of real-time processing of events and streams and the investigation of relevant support services for dataspaces.
- *Discovery:* A key challenge within a dataspace is to locate relevant participants in the dataspace, identify relationships among participants, and improve the understanding of existing relationships among participants. [2, 107, 112, 113].
- *Reusing Human Attention:* The primary focus of research in this area looks to capture "user attention" to support management in the dataspace [114–117]. Leveraging user's attention for improving integration in dataspaces is considered an integral part of any dataspace application or platform [2, 78]. Roomba [118] exploits user feedback for improving integration of dataspaces using a decision-theoretic technique to quantify the desirability state

of a dataspace. DSToolkit [81, 119] uses end-user feedback for annotating, selecting and refining schema mappings, and for estimating the precision and recall of query results in a dataspace. MOBS [120] focuses on collecting feedback from many users to make a decision based on the combined result.

- *Storage and Indexing:* A key challenge for dataspaces is dealing with the heterogeneity of storage and indexing mechanisms of the various participants to create a uniformly indexed dataspace [2, 121]. Creating local storage and the placement of data and indices for optimal performance within a wide-area deployment are also active areas of research [122].
- *Correctness Guarantees:* Enabling access to a set of disparate data sources with confidence is a crucial challenge for dataspaces. To achieve this, there are a number of challenges in the quality of results, the effects and permanence of updates, and varying levels of service [2, 123], in a heterogeneous, highly autonomous environment.
- *Theoretical Foundations:* The concept of dataspaces opens several questions on their theoretical foundation. There is a need for research on the formal understanding of data governance [1], data models, relations among participants, and queries in a dataspace [2, 81].

## 3.9 Summary

In this chapter, we explored the challenges associated with data management and integration in the era of big data. Within large-scale integration scenarios that involve thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources due to the challenge posed by the long tail of data variety. We detailed the dataspace paradigm as an emerging data management approach that embraces the notion of "good enough" and best-effort approximations as a means of data management. Data is integrated on an "as-needed" basis with the labour-intensive aspects of data integration postponed until they are required. Dataspaces reduce the initial effort required to set up data integration by relying on automatic matching and mapping generation techniques. When tighter semantic integration is required, it can be achieved in an incremental "pay-as-you-go" fashion by detailed mappings among the required data sources. This chapter detailed the fundamentals of the dataspace paradigm, including their core principles, comparison to existing approaches, support platform, support services, life cycle, and research challenges.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

# Chapter 4
# Fundamentals of Real-time Linked Dataspaces

**Keywords** Data platform · Linked data · Stream processing · Event processing · Dataspaces · Internet of Things · Incremental data management

## 4.1 Introduction

Dataspaces can provide an approach to enable data management in smart environments that can help to overcome technical, conceptual, and social/organisational barriers to information sharing. However, there has been limited work on the use of dataspaces within smart environments and the necessary support services for real-time events and data streams. This chapter introduces the Real-time Linked Dataspace (RLD) as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data, knowledge graphs, and (near) real-time processing capabilities. The RLD has been specifically designed to support the sharing and processing of data between intelligent systems within smart environments. We propose a set of specialised dataspace support services to enable the requirements of loose administrative proximity and semantic integration for event and stream systems. These requirements form the foundation of the techniques and models used to process events and streams within the RLD.

The chapter is structured as follows: event and stream processing for the Internet of Things are discussed in Sect. 4.2, and the fundamentals of RLD are defined in Sect. 4.3 including principles, comparison to existing dataspaces, and the main components of the architecture. Section 4.4 details a principled approach to pay-as-you-go data management and introduces the 5 star pay-as-you-go model for RLD support services. Section 4.5 introduces the support platform for the RLD, Sect. 4.6 discusses its suitability as a data platform for intelligent systems within smart environments by comparison to similar platforms, and a summary is provided in Sect. 4.7.

© The Author(s) 2020
E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_4

## 4.2 Event and Stream Processing for the Internet of Things

At the end of the twentieth century and in the first decade of the twenty-first century, a recognition emerged among researchers and practitioners that a new class of information processing systems was needed. The need for the event processing paradigm emerged from a range of diverse distributed applications that required on-the-fly and low-latency processing of information items. Example applications include environmental monitoring [124], stock market analysis [125], RFID-based inventories [126], resource management (e.g. energy, water, mobility) in smart environments [4], and security systems such as intrusion detection [127].

Smart environments have emerged in the form of smart cities, smart buildings, smart energy, smart water, and smart mobility, all of which have large quantities of real-time data that must be processed. The Internet of Things (IoT) is producing events and streams that are generating significant quantities of data within smart environments which are driving a new wave of data-driven intelligent systems that can more effectively and efficiently manage resources while also providing enhanced user experience.

The paradigms of event processing and stream processing have evolved through the work of several communities, including active databases [128, 129], reactive middleware [130–132], event-based software engineering [133, 134], and Message-Oriented Middleware [135]. As these paradigms emerged, they created their own communities around data stream management systems [136–139], event processing systems [140, 141], Complex Event Processing [140], and Publish-Subscribe [142, 143].

Cugola and Margara [139] complement this picture to justify a new umbrella paradigm for a set of emerging systems where "timeliness and flow processing are crucial" and call it the Information Flow Processing (IFP) Domain. Figure 4.1 presents an elaboration of Cugola and Margarita's functional model and shows the main functionalities of an IFP engine for single and complex event processing. In event processing, data items that are shared in real-time are called *events*. An event

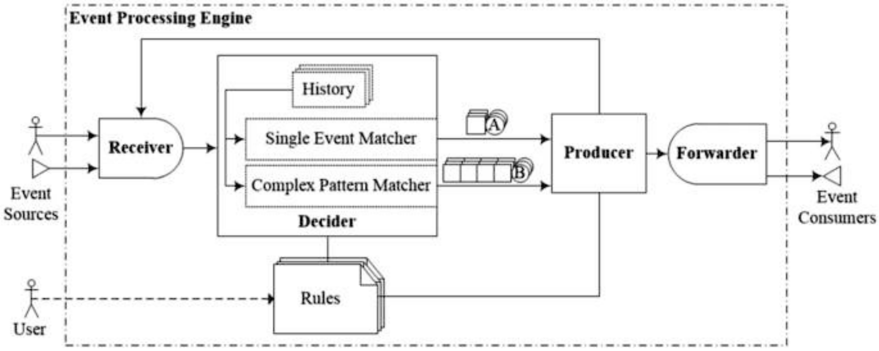


Fig. 4.1 The information flow processing model for single and complex event processing [144]

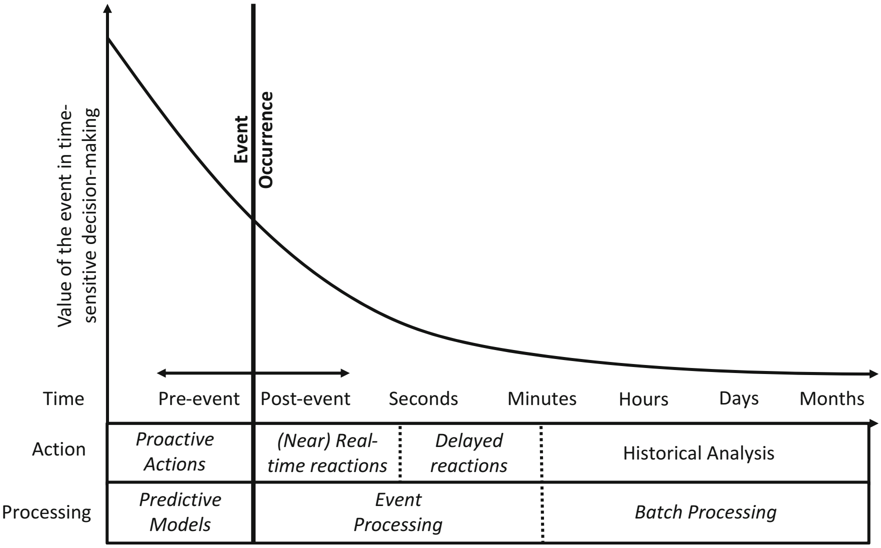


**Fig. 4.2** The value of events in time-sensitive decision-making, actions, and processing approaches

can take the form of a sensor reading. Data sources which produce events are called *event producers*. Users and software which are interested in an event, or set of events, are called *event consumers*. An essential part of the event processing paradigm is the matching mechanism between the events and the interests of event consumers. This is similar to the concept of query processing in relational database management systems, where events replace the concept of a data tuple, and subscriptions or rules replace the concept of queries. In a specific family of event processing, called stream processing, queries take the name of continuous queries as they are evaluated continuously against moving data.

### *4.2.1 Timeliness and Real-time Processing*

The concept of timeliness has been expressed in the literature using various terms such as low latency [145, 146], high throughput [146, 147], low delay [148], and real-time processing [3, 149]. All these terms, except real-time processing, can be classified under the umbrella of fast computing. This term means that the system is efficient in processing information items in a way that the ratio Value/Time is maximised as detailed in Fig. 4.2.

Another perspective is that real-time processing includes the notion of executing the information processing task within a time constraint, called a deadline [150]. Timeliness, as described by Cugola and Margara [139], is more similar to the concept of fast computing. Technically, it can be measured by the related concepts of latency and throughput:

- *Latency* is defined in this book as the total time required to process an information item starting from its arrival to the processing agent until its completion.
- *Throughput* is the number of information items wholly processed within a time unit.

Thus, real-time, whenever used in this book, means low latency, high throughput, and processing as soon as the information items are available. Near-real-time processing, when used, should also be taken to have this meaning. The ability to quickly react to an event is a critical requirement within many real-world situations. Systems that can provide data processing capabilities suitable for real-time information need to be designed in a specific manner, Stonebraker et al. [3] suggest eight requirements for an effective and efficient design:

- *Rule 1—Keep the Data Moving:* Process messages "in-stream" without the need to store the message to perform an operation or sequence of operations.
- *Rule 2—Query Using SQL on Streams:* Support a high-level stream language with built-in extensible stream-oriented primitives and operators.
- *Rule 3—Handle Stream Imperfections:* Built-in mechanisms to provide resiliency against stream "imperfections" including delayed, missing, and out-of-order data which occur in real-world data streams.
- *Rule 4—Generate Predictable Outcomes:* Guarantee predictable and repeatable outcomes.
- *Rule 5—Integrate Stored and Streaming Data:* Ability to efficiently store, access, and modify state information, and combine it with live streaming data.
- *Rule 6—Guarantee Data Safety and Availability:* Provide high availability and integrity of the data maintained.
- *Rule 7—Partition and Scale Applications Automatically:* Ability to distribute processing across multiple computing resources for incremental scalability.
- *Rule 8—Process and Respond Instantaneously:* Highly optimised, minimal-overhead execution engine to deliver a real-time response.

## 4.3 Fundamentals of Real-time Linked Dataspaces

Real-time data sources are increasingly forming a significant portion of the data generated in the world. This is in part due to increased adoption of the Internet of Things and the use of sensors for improving data collection and monitoring of smart environments, which enhance different aspects of our daily activities in smart buildings, smart energy, smart cities, and others [1]. To support the interconnection of intelligent systems in the data ecosystem that surrounds a smart environment, there is a need to enable the sharing of knowledge among systems. A data platform can provide a clear framework to support the sharing of data among a group of intelligent systems within a smart environment [1] (see Chap. 2). In this book, we advocate the use of the dataspace paradigm within the design of data platforms to enable data ecosystems for intelligent systems.

A dataspace is an emerging approach to data management which recognises that in large-scale integration scenarios, involving thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources [2]. Within dataspaces, datasets *co-exist* but are not necessarily fully integrated or homogeneous in their schematics and semantics. Instead, data is integrated on an "*as-needed*" basis with the labour-intensive aspects of data integration postponed until they are required. Dataspaces reduce the initial effort required to setup data integration by relying on automatic matching and mapping generation techniques. This results in a loosely integrated set of data sources. When tighter semantic integration is required, it can be achieved in an incremental "*pay-as-you-go*" fashion with more detailed mappings among the required data sources.

Within the dataspace paradigm, there has been limited work on addressing the requirements of real-time processing of events and streams, and research into relevant support services. The Real-time Linked Dataspace (RLD) has been created as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data, knowledge graphs, and real-time stream and event processing capabilities to support large-scale distributed heterogeneous collection of streams, events, and data sources [4]. This work builds on past efforts to use dataspaces in Building Data Management [62], Energy Data Management [100], and System of Systems [85]. The goal is to support a principled approach to incremental real-time data management based on a set of support services with tiered levels of support, to provide a unified entity-centric query framework over real-time and historical data streams in a smart environment.

This section details the foundations of the Real-time Linked Dataspace approach and describes how its architectural components meet the key requirements identified for real-time information processing (as identified by Stonebraker et al. [3]) and data platform for smart environments (as identified in Chap. 2).

### 4.3.1 Foundations

The Knowledge Value Ecosystem (KVE) framework (see Chap. 2) helps us to understand the challenge with knowledge flows at different levels within data ecosystems. At the knowledge exchange level, a data platform for a smart environment would need to overcome administrative and semantic barriers. Acknowledging that the core challenges within dataspaces are lack of administrative proximity, and loose semantic integration, the question becomes: how can we better tailor the principles of dataspaces, to real-time data processing? To answer this question, we look no further than the literature of the event processing paradigm itself. A core principle in event processing is decoupling, which refers to the lack of explicit agreements in order to increase scalability as defined by Eugster et al. [142]. Three main dimensions have been recognised in the event processing literature with respect to decoupling:

- *Space Decoupling*, which means that event producers and consumers do not hold identifiers (e.g. IP addresses) of each other.
- *Time Decoupling*, which means that event producers and consumers do not have to be active at the same time.
- *Synchronisation Decoupling*, which means that event producers and consumers do not block each other when exchanging events.

These dimensions can be classified within the "administrative proximity" aspect of data management from Franklin et al. [2]. The decoupled nature of event-based systems reduces their administrative proximity. However, in terms of semantic integration, event-based systems currently require tight semantic integration. Hasan and Curry [151] identified forms of semantic integration based on Carlile's framework within event-based systems:

- *Syntactic Coupling:* The amount of agreement among participants in the event processing environment on the sharing and establishment of a common low-level syntax. This view has been established by Shannon and Weaver [38] in their communication theory, where syntax has the form of zeros and ones. They claim that once such a syntax is shared, accurate communication can be ensured, and the task becomes that of information processing rather than communication.
- *Semantic Coupling:* The amount of agreement among participants in the event processing environment on the mappings among symbols used in event messages and the meanings to which they refer.

Bringing both of these views together helps us to understand the challenge with knowledge flows within data ecosystems. Here we build on the concept of decoupling to meet the principles of dataspaces by Halevy et al. [78], as illustrated in Fig. 4.3. Here we can see that within an RLD the main administrative issues are around space, time, and synchronisation of interacting systems. While semantic integration is centred on syntactic and semantic concerns, this requires a Real-Time Linked Dataspace to support an event processing paradigm that supports many formats of data, does not depend on schema agreement, and supports a best-effort approximate and pay-as-you-go approach.

### 4.3.2 Definition and Principles

A Real-time Linked Dataspace is a specialised dataspace that manages and processes the large-scale distributed heterogeneous collection of streams, events, and data sources [4]. It manages the sources without presuming a pre-existing semantic integration among them, uses linked data and knowledge graphs to coordinate the dataspace, and operates under a 5 star model for "pay-as-you-go" data management (see Sect. 4.4).

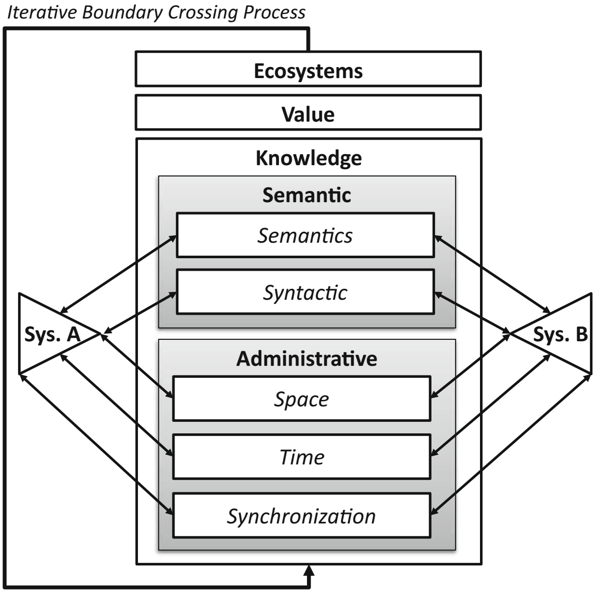


**Fig. 4.3** Dimensions of decoupling for knowledge flows between event-based systems based on the KVE framework

The RLD adapts the dataspace principles as set out by Halevy et al. [78] to describe the specific requirements within a real-time dataspace setting:

- A Real-time Linked Dataspace must deal with many different formats of streams and events.
- A Real-time Linked Dataspace does not subsume the stream and event processing engines; they still provide individual access via their native interfaces.
- Queries in the Real-time Linked Dataspace are provided on a best-effort and approximate basis.
- The Real-time Linked Dataspace must provide pathways to improve the integration among the data sources, including streams and events, in a pay-as-you-go fashion.

In order to enable these principles to support real-time data processing, we propose a set of specialised dataspace support services to enable the requirements of loose administrative proximity and semantic integration for event and stream systems. Loose coupling of event processing systems on the semantic dimension reflects a low cost to define and maintain rules concerning the use of terms, and a low cost to building and agreeing on the event semantic model. This requirement forms the foundation of the techniques and models used to process events and streams within the Real-time Linked Dataspace.

**Table 4.1** Comparison of DBMS, Dataspace, and a Real-time Linked Dataspace

| | DBMS | Dataspace | Real-time Linked Dataspace |
|---|---|---|---|
| Data management requirements | | | |
| Model | Relational | All | All |
| Formats | Homogeneous | Heterogeneous | Heterogeneous |
| Schema | Schema first, data later | Data first, schema later or never | Data first, schema later or never |
| Control | Complete | Partial | Partial |
| Leadership | Top-down | Top-down/Bottom-up | Distributed |
| Query | Exact | Approximate | Approximate |
| Integration | Upfront | Incremental | Incremental |
| Architecture | Centralised | Decentralised | Distributed |
| Data processing | None | None | Real-time and batch processing |
| Real-time requirements | | | |
| Rule 1: Keep the data moving | No | Possible | Yes |
| Rule 2: High-level stream language | No | Possible | Yes |
| Rule 3: Handle stream imperfections | Difficult | Possible | Yes |
| Rule 4: Predictable outcome | Difficult | Possible | Possible |
| Rule 5: High availability | Possible | Possible | Possible |
| Rule 6: Stored and streamed data | No | Possible | Yes |
| Rule 7: Distribution and scalability | Possible | Possible | Possible |
| Rule 8: Instantaneous response | Possible | Possible | Yes |

### *4.3.3 Comparison*

While the initial vision of dataspaces encompassed the notion of support for data streams, the details of how to specifically handle streams within a dataspace were not covered in depth. The RLD goes beyond a traditional dataspace approach [2] by supporting the management of entities within the dataspace as first-class citizens along with data sources, and it extends the dataspace support platform with real-time processing and querying capabilities for streams and events as detailed in Table 4.1.

### *4.3.4 Architecture*

The RLD contains all the relevant information within a data ecosystem, including things, sensors, and data sources and has the responsibility for managing the relationships among these participants. Figure 4.4 illustrates the architecture of the RLD with the following main concepts:

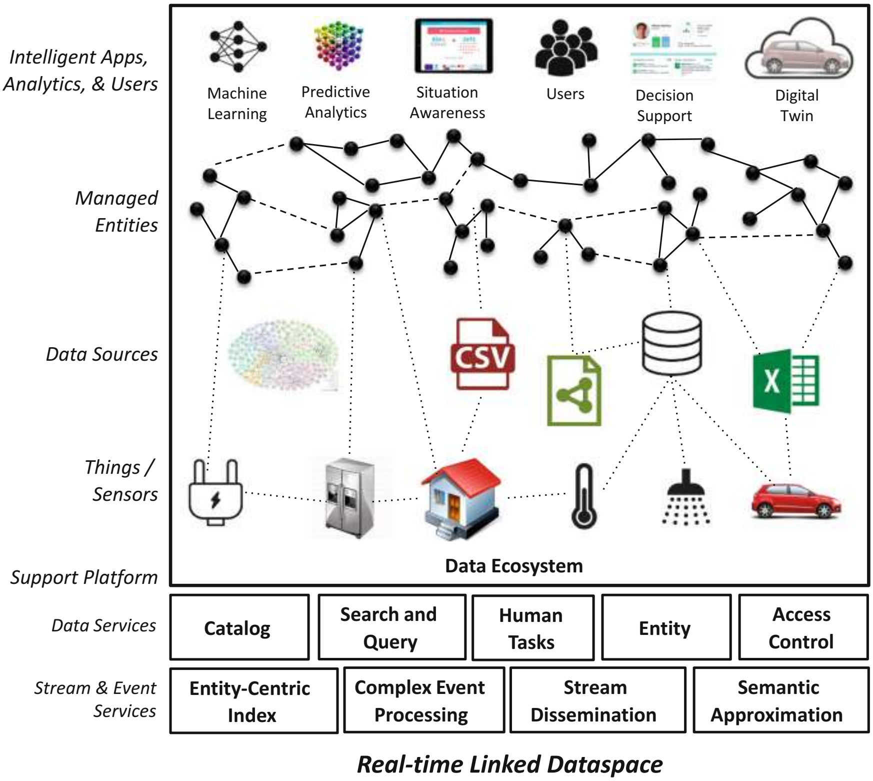


**Fig. 4.4** Real-time Linked Dataspace architecture

- *Support Platform:* Responsible for providing the functionalities and services essential for managing the dataspace. Support services are grouped into data services and stream and event services.
- *Things/Sensors:* Produce real-time data streams that need to be processed and managed. Things in a smart environment range from connected devices, energy, and water sensors, to connected cars and manufacturing equipment.
- *Data Sources:* Available in a wide variety of formats and accessible through different systems interfaces. Example data sources include building management systems, energy and water management systems, passenger information systems, financial data, weather, and (linked) open datasets.
- *Managed Entities:* Actively managed entities (e.g. people, equipment, buildings) within the data ecosystem, including their relationship to participating things, data sources, and other entities in the RLD.
- *Intelligent Applications, Analytics, and Users:* Interact with the RLD and leverage its data and services to provide data analytics, decision support tools, user interfaces, and data visualisations. Applications/Users can query the RLD in an entity-centric manner, while users can be enlisted in the curation of the data and entities via the Human Task service.

## 4.4 A Principled Approach to Pay-As-You-Go Data Management

Within the RLD, the pay-as-you-go approach to data management is complemented with a principled tiered approach to the design of support services where an increase in the level of active data management has a corresponding increase in the associated effort [4]. This tiered approach to data management provides flexibility by reducing the initial effort and barriers to joining the dataspace. The tiers for the RLD are a specialisation of the 5 stars scheme defined by Tim Berners-Lee for publishing open data on the web [43].

### *4.4.1 TBL's 5 Star Data*

The W3C Linking Open Data (LOD) project started in 2007 and began publishing datasets under open licenses and following the linked data principles. To encourage people to publish linked data, the inventor of the web and the initiator of the linked data paradigm, Tim Berners-Lee, proposed a 5 star rating system [41]. The rating system (see Fig. 4.5) helps data publishers to evaluate how much their datasets conform to the linked data principles. The first star is to make data available on the web, with each additional star corresponding to increased reusability and interoperability of the published data as more of the principles of linked data are followed.

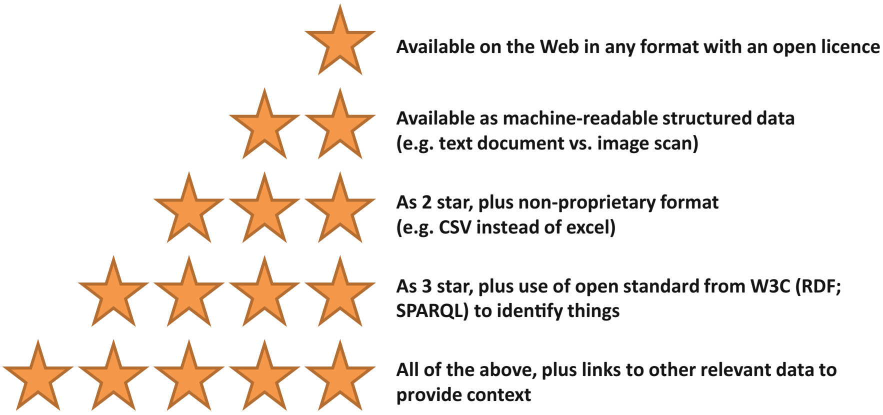

Available on the Web in any format with an open licence
Available as machine-readable structured data (e.g. text document vs. image scan)
As 2 star, plus non-proprietary format (e.g. CSV instead of excel)
As 3 star, plus use of open standard from W3C (RDF; SPARQL) to identify things
All of the above, plus links to other relevant data to provide context

**Fig. 4.5** Tim Berners-Lee's 5 star rating system [41]

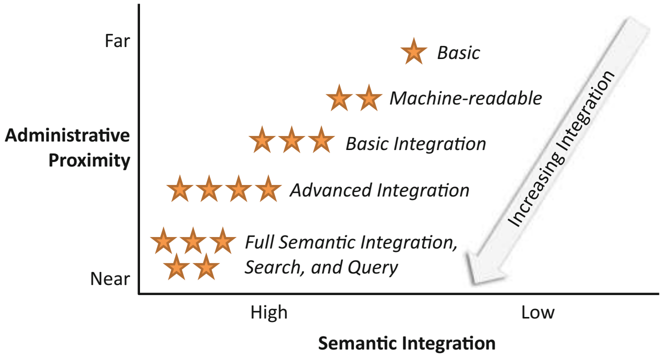


**Fig. 4.6** The 5 star pay-as-you-go model of a Real-time Linked Dataspace

### *4.4.2 5 Star Pay-As-You-Go Model for Dataspace Services*

In contrast to the classical one-time integration of datasets that causes a significant upfront overhead, the RLD adopts a principled pay-as-you-go paradigm for supporting an incremental approach to data management. At the foundation of the approach is the principle that the publisher of the data is responsible for paying the cost of joining the dataspace. This pragmatic decision allows the RLD to grow and enhance gradually with participants joining or leaving the dataspace at any time. The next principle is that data is managed following a tiered approach, where an increase in the level of active data management has a corresponding increase in associated costs.

The tiered approach to data management provides flexibility by reducing the initial cost and barriers to joining the dataspace. The tiers are described using a specialisation of the 5 star scheme defined by Tim Berners-Lee. The original star scheme has been extended to consider the level of integration of the data sources with the support services of a dataspace. At the minimum level, a data source needs to be made available with a dataspace. Over time, the level of integration with the support services can be improved in an incremental manner on an as-needed basis. The more the investment made to integrate with the support services, the better is the integration achievable in the dataspace. The 5 star pay-as-you-go model for the RLD is illustrated in Fig. 4.6. The five tiers are:

*1 Star* **Basic:** The participant is published in the dataspace with limited or no integration with support services.

*2 Stars* **Machine-readable:** The participant is publishing data in a machine-readable format. This enables services to provide a minimal level of support with basic functionality (e.g. browsing) where available basic interfaces are exposed.

*3 Stars* **Basic integration:** The use of a non-proprietary format enables support services to provide essential services at the data-item/entity level with support for simple functionality (e.g. keyword search).

*4 Stars* **Advanced integration:** The participant is integrated with most support service features (e.g. structured queries) with an awareness of its relationships to other participants with basic support for federation.

*5 Stars* **Full semantic integration, search, and query:** The participant is fully integrated into the support services (e.g. question answering) and linked to relevant participants. It plays its full role in the global view of the dataspace.

## 4.5 Support Platform

The RLD-Support Platform (RLD-SP) provides a set of core services to support intelligent application developers and data scientists with a base functionality when working with sources in the RLD. Each of the services in the RLD-SP has been designed to follow the pay-as-you-go paradigm to support varying levels of service offerings to the participants in the dataspace. Two categories of support services have been developed, one targeting core data management and the other focuses on support for streams and events. This section details these services and their tiered levels of support as detailed in Table 4.2.

### *4.5.1 Data Services*

The RLD-SP provides a set of enhanced data support services to enable all participants in the dataspace to get setup and running with a low overhead. These support services are built on the core support services defined by Franklin et al. and extended to follow the entity-centric data management approach of knowledge graphs [2]. The services have been designed to include support for linked data and follow the 5 star pay-as-you-go model. Examples of data services include the catalog, entity management, search and query, and data service discovery. Part II of this book further explores these data services.

### *4.5.2 Stream and Event Processing Services*

The RLD supports real-time data processing with support services that follow the data management philosophy of dataspaces. The RLD-SP provides support services to handle the processing of streaming and event data tackling issues including entity-centric queries, complex event processing, stream dissemination, and semantic approximation. The goal of these services is to support participants in the RLD to get setup and running with a low overhead for administrative setup costs (e.g. establishing data agreements, service selection, and composition). Part III of this book further explores the stream and event support services.

**Table 4.2** The 5 star pay-as-you-go scheme for the Real-time Linked Dataspace support services [4]

| Pay-as-you-go star rating | Data format | Catalog | Access control | Search and query | Entity | Human tasks | Entity-centric index | Complex event processing | Stream dissemination | Semantic approximation |
|---|---|---|---|---|---|---|---|---|---|---|
| *<br>Basic | Any format (e.g. PDF) | Registry of datasets and streams | None | Browsing | None | None | None | None | None | None |
| **<br>Machine-readable | Machine-readable structured data (e.g. Excel) | Non-machine-readable metadata document (e.g. PDF) | Coarse-grained (Dataset level) | Keyword search | Entities identifiers in documentation | Schema mapping | Basic processing | Single stream | None | Semantic matching |
| ***<br>Basic integration | Non-proprietary format (e.g. CSV, JSON, XML) | Machine-readable metadata<br>Equivalence among schema concepts | Fine-grained (Entity-level)<br>Secure query service | Structure search | Source level (siloed) | Entity mapping | Historical views of streams | Multi-service composition | Point-to-point | Thematic matching |
| ****<br>Advanced integration | Uses the first two principles of linked data | Relations among schemas (dataspace level) | Data anonymisation | Structured queries | Canonical identifiers and entity mappings across sources | Entity enrichment | Stream enrichment with context and entity data | Quality of service aware service composition | Wireless broadcast | Entity-centric matching |
| *****<br>Full semantic integration, search, and query | Follows all publishing principles of linked data | Full semantic mappings | Usage control | Schema-agnostic question answering | Knowledge graphs semantically link entities to related entities, data, and streams | Data quality improvement | Entity-centric real-time query | Context-aware | Complex patterns | Context-aware |

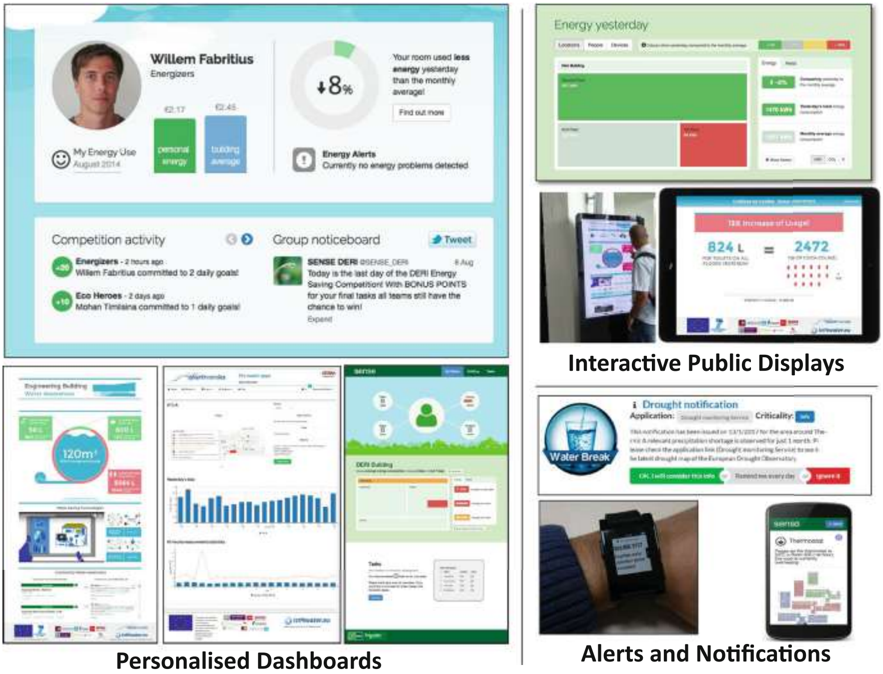

Willem Fabritius
Energizers
My Energy Use
August 2014
€2.17 personal energy
€2.45 building average
↓8%
Your room used **less energy** yesterday than the monthly average!
Find out more
**Energy Alerts**
Currently no energy problems detected
Competition activity
**Energizers** - 2 hours ago
Willem Fabritius committed to 2 daily goals!
**Eco Heroes** - 2 days ago
Mohan Timilsina committed to 1 daily goals!
Group noticeboard
Tweet
**SENSE DERI** @SENSE_DERI 8 Aug
Today is the last day of the DERI Energy Saving Competition! With BONUS POINTS for your final tasks all teams still have the chance to win!
Expand
Energy yesterday
Drought notification
Application:
Criticality:

**Fig. 4.7** Intelligent systems and applications built using the Real-time Linked Dataspace [16]

## 4.6 Suitability as a Data Platform for Intelligent Systems Within IoT-Based Smart Environments

The RLD has been used as a data platform to support the development of intelligent systems and applications within a range of smart environments including smart home, school, office building, university, and airport [16]. Within these environments, a data platform needs to support a wide range of end-users with different interests and priorities from corporate managers looking for data to improve the performance of their business to software engineers developing applications for the smart environment (see Fig. 4.7). Intelligent systems and applications developed using the RLD are discussed in detail in part IV of this book. In this section, we examine the suitability of the RLD as a data platform for a smart environment.

### 4.6.1 Common Data Platform Requirements

The goal of the RLD is to support a principled approach to incremental real-time data management based on a set of services with tiered levels of support within a smart environment. The FAIR principles (Findability, Accessibility, Interoperability, and Reusability) are designed to enable good data management to support knowledge discovery and innovation, and the subsequent data and knowledge integration and reuse [61]. The principles are described further in Chap. 2. Within the context of this work, the principles can serve as a high-level guide for the design of a data platform to support knowledge sharing within a smart environment.

Section 2.4 identified a set of common data management requirements for a data platform. The common data platform requirements are [4]:

- *Standard Data Syntax, Semantics, and Linkage:* Facilitate integration and sharing, ideally with open standards and non-proprietary approaches.
- *Single-Point Data Discoverability and Accessibility:* Allow the organisation and access to datasets and metadata through a single location.
- *Incremental Data Management:* Enable a low barrier to entry and a pay-as-you-go paradigm to minimise costs.
- *Secure Access Control*: Support data access rights to preserve the security of data and privacy of users in the smart environment. Access control is needed at both the level of the data source and at the level of the data item (i.e. entity-value).
- *Real-time Data Processing:* Including ingestion, aggregation, and pattern detection within event streams originating from sensors and things in the smart environment.
- *Unified Querying of Real-time Data and Historical Data:* Provide applications and end-users with a holistic queryable state of the smart environment at a latency suitable for user interaction.
- *Entity Management:* The storage, linkage, curation, and retrieval of entity data, such as users, zones, and locations.
- *Event Enrichment:* Enhancement of sensor/things streams with contextual data (e.g. entities) to make the stream data more encapsulated and useful in downstream processing.

These requirements can be used to survey the capabilities of existing approaches for data platforms within a smart environment and to highlight the main contribution of the RLD.

### 4.6.2 Related Work

The CityPulse [65] project provides a distributed system for semantic discovery, data analytics, and interpretation of large-scale and near-real-time Internet of Things data and social media data streams [14]. In addition to providing unified views of the data, CityPulse also provides data analytics modules that perform intelligent data aggregation, event detection, quality assessment, contextual filtering, and decision support. CityPulse supports open standards for semantics, real-time stream processing, and entity management. However, no support exists for single-point data access, a pay-as-you-go data management paradigm, unified views over real-time and historical data, security, and event streams enrichment.

The OpenIoT [66] platform enables the semantic interoperability of IoT services in the cloud through the use of the W3C Semantic Sensor Networks (SSN) ontology [152], which provides a common standards-based model for representing physical and virtual sensors. OpenIoT provides middleware for uniform access to IoT data and support for the development and deployment of IoT applications. OpenIoT supports open standards for semantics, real-time stream processing, security, and entity management. However, it lacks support for single-point data access, a pay-as-you-go data management paradigm, unified views over live and historical data, and event streams enrichment.

The SmartSantander project developed the City Data and Analytics Platform (CiDAP) [33], a centralised platform to access data generated from multiple heterogeneous sensors installed in a city. The platform can deal with historical data and near real-time information in an architecture like Lambda. CiDAP provides limited support for data management beyond the low-level sensor streams and pushes these concerns to the application-level. The result is applications duplicating common data management functionalities. SmartSantander follows open standards for semantics, single-point data access, security, real-time stream processing, and partial unified queries over streams and datasets. However, it lacks support for an incremental data management paradigm, entity management, or event streams enrichment.

The Spitfire [64] project uses semantic technologies to provide a uniform way to search, interpret, and transform sensor data. Spitfire works towards a Semantic Web of Things, by providing abstractions for things, basic services for search and annotation, as well as by integrating sensors and things into the LOD cloud. Spitfire adopts semantic web standards for describing data, partial secure access control, entity management, and event enrichment. It does not support single-point access for data, incremental data management, real-time data processing, or unified queries for real-time and legacy data.

ThingStore [153] provides a “marketplace” for IoT applications development with the ability to deploy and host them. The platform provides support for event detection, service discovery, an Event Query Language, together with event notification and management. The architecture of ThingStore is a computation hub to connect things, software, and end-users. ThingStore supports secure and real-time data processing. However, it lacks support for open standards to describe data, single-point access for data, entity management and event enrichment, incremental data management, and unified queries for real-time and legacy data.

From the analysis in Table 4.3, we note that existing data platforms support semantic descriptions of data according to open standards such as semantic web and linked data. However, they lack an incremental data management paradigm and do

**Table 4.3** Comparison of related frameworks to common data platform requirements [4]

| Requirements | City pulse [65] | Open IOT [66] | SmartSantander [33] | Spitfire [64] | ThingStore [153] | Real-time Linked Dataspace |
|---|---|---|---|---|---|---|
| Standard data syntax, semantics, and linkage | Yes | Yes | Partial | Yes | No | Yes |
| Single-point data discoverability and accessibility | No | No | Partial | No | No | Yes |
| Incremental data management | No | No | No | No | No | Yes |
| Secure access control | No | Yes | Yes | Partial | Partial | Yes |
| Real-time data processing | Yes | Yes | Yes | No | Yes | Yes |
| Unified querying of real-time data and historical data | No | No | Partial | No | No | Yes |
| Entity management | Partial | Yes | No | Yes | No | Yes |
| Event enrichment | No | No | No | Partial | No | Yes |

not support a single access point to discover and access datasets. Most data platforms address the real-time processing of data but do not provide unified access to it along with historical data. Half of the data platforms provide some support for entity management. However, streams are not typically enriched with contextual data. Based on the analysis of the requirements identified we could see there is a clear need for an incremental pay-as-you-go data management, a single point of data/stream access, support for entity-centric views of real-time and historical data, and streams enrichment for better entity-centric and contextual data retrieval.

## 4.7 Summary

Real-time Linked Dataspaces (RLD) enable data ecosystems for intelligent systems within Internet of Things-based smart environments by providing a principled approach to the incremental management of stream events that can reduce the technical and conceptual barriers to information sharing. This chapter introduces the Real-time Linked Dataspace that combines the pay-as-you-go paradigm of dataspaces and linked data with real-time search and query capabilities. The chapter details the fundamentals of the RLD and its basis in the stream and event processing, and dataspace communities. The design of the RLD is detailed, including the main components of the architecture, the 5 star model for pay-as-you-go support services including services for streams, events, and data. Finally, a high-level overview of the suitability of the RLD for a data platform is provided with a comparison to related platforms.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

# Part II
# Data Support Services

Part II of this book explores the critical data management support services within Real-time Linked Dataspaces, including catalogs, entity management, query and search, data service discovery, and human-in-the-loop tasks.

# Chapter 5
# Data Support Services for Real-time Linked Dataspaces

**Keywords** Dataspaces · Incremental data management · Data services

## 5.1 Introduction

The objective of a Real-time Linked Dataspace is to support a real-time response from intelligent systems to situations of interest when a set of events take place within a smart environment. In addition to the obvious need for real-time data processing support services, there is also the need for the fundamental data support services one would expect in a dataspace support platform. This part of the book discusses the enhanced data support services developed for the Real-time Linked Dataspace to support data management for intelligent systems within smart environments. The goal of these services is to support a real-time dataspace system to get up and running with a low overhead for administrative setup costs (e.g. catalog, entity management, search and query, and data service discovery). Each of the support services has been specifically designed for and evaluated within Internet of Things-based smart environments. This chapter provides a high-level overview of the data support services discussed in Part II and details their tiered service levels. This chapter is structured as follows: Sect. 5.2 provides a brief overview of the pay-as-you-go data support services covered in this part of the book, while Sect. 5.3 details how the services support the 5 star scheme. A summary is provided in Sect. 5.4.

## 5.2 Pay-As-You-Go Data Support Services for Real-time Linked Dataspaces

The adoption of the Internet of Things (IoT)-enabled smart environments is empowering data-driven systems that are transforming our everyday world. To support the interconnection of intelligent systems in the data ecosystem that

© The Author(s) 2020

E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_5

surrounds a smart environment, there is a need to enable the sharing of data among intelligent systems. A data platform can provide a clear framework to support the sharing of data among a group of intelligent systems within a smart environment [1] (see Chap. 2). In this book, we advocate the use of the dataspace paradigm within the design of data platforms to enable data ecosystems for intelligent systems.

We have created the Real-time Linked Dataspace (RLD) (see Chap. 4) as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data, knowledge graphs, and real-time stream and event processing capabilities to support large-scale distributed heterogeneous collection of streams, events, and data sources [4]. At the foundation of the pay-as-you-go approach to data integration is the idea that the owners of the data sources are responsible for the incremental improvement in the integration and quality of data available in the dataspace. The needs of the user drive incremental improvements over time. This pragmatic approach allows the dataspace to grow and enhance gradually with data sources or streams joining or leaving at any time. In order to reduce the burden on data source owners and users of the RLD, a support platform with a number of data support services is provided.

The design of the support services needs to conform to the principles of RLDs. The RLD principles specialise the dataspace principles as set out by Halevy et al. [78] to describe the specific requirements within a real-time dataspace setting:

- A Real-time Linked Dataspace must deal with many different formats of streams and events.
- A Real-time Linked Dataspace does not subsume the stream and event processing engines; they still provide individual access via their native interfaces.
- Queries in the Real-time Linked Dataspace are provided on a best-effort and approximate basis.
- The Real-time Linked Dataspace must provide pathways to improve the integration among the data sources, including streams and events, in a pay-as-you-go fashion.

The data services provided by RLD (Fig. 5.1) are:

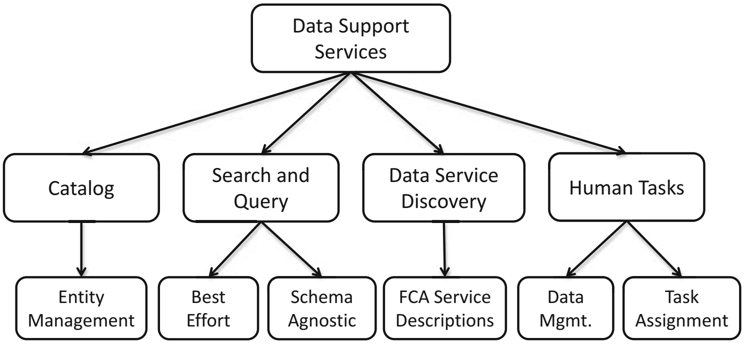


**Fig. 5.1** Data services provided by the real-time linked dataspace

- *Catalog:* The catalog service plays a crucial role by providing information about participating data sources in the dataspace. Within the catalog, all datasets and entities are declared along with relevant metadata.
- *Entity Management:* The Entity Management Service (EMS) manages information about the entities (e.g. real-world objects) in the dataspace. The EMS is an essential service for decision-making applications that rely on accurate entity information.
- *Search and Query:* The Search and Query services help developers, data scientists, and users to find relevant data sources within the dataspace.
- *Data Service Discovery:* Efficiently describing and organising data sources in dataspaces is essential. The Data Service Discovery Service organises and indexes data sources based on their capabilities.
- *Human Tasks:* The Human Task service is concerned with the collaborative aspect of data management within the dataspace by enabling small data management tasks (e.g. data quality and enrichment) to be distributed among users in the smart environment. The Human Task service can also engage participants in citizen actuation tasks within the smart environment.

An essential requirement for intelligent systems within a smart environment is to support the querying of real-time data streams. Within the RLD this is achieved by several support services for processing streams and events which are covered in Part III of this book. In the remainder of this part of the book, we detail the above data support services and focus on how they enable data management in the RLD. Each of these services has been designed to follow the RLD principles and to offer tiered service-levels following the 5 star pay-as-you-go model from Chap. 3.

## 5.3 5 Star Pay-As-You-Go Levels for Data Services

The 5 star scheme details the level of integration of the data sources with the support services of a dataspace. At the 1 star level, a data source needs to be made available with the dataspace. Over time, the level of integration with the support services can be improved in an incremental manner on an as-needed basis. The more the investment is made to integrate with the support services, the better is the integration achievable in the dataspace. The different service tiers of the RLD data support services are detailed in Table 5.1.

## 5.4 Summary

This chapter provides an overview of the enhanced data support services developed for the Real-time Linked Dataspace to enable intelligent systems within IoT-based smart environments. The goal of these services is to support Real-time Linked

**Table 5.1** The 5 star pay-as-you-go model for the Real-time Linked Dataspace Support Services [4]

| Pay-as-you-go star rating | Data format | Catalog | Access control | Search and query | Entity | Human tasks |
|---|---|---|---|---|---|---|
| *<br>Basic | Any format (e.g. PDF). | Registry of datasets, and streams | None | Browsing | None | None |
| **<br>Machine-readable | Machine-readable structured data (e.g. Excel)<br>Documentation to understand the data/stream structure, format, and characteristics. | Non-machine-readable metadata document (e.g. PDF) | Coarse-grained (Dataset level) | Keyword search | Entities identifiers in documentation | Schema mapping |
| ***<br>Basic integration | Non-proprietary format (e.g. CSV, JSON, XML) | Machine-readable metadata<br>Equivalence among schema concepts | Fine-grained (Entity-level)<br>Secure query service | Structure search | Source level (siloed) | Entity mapping |
| ****<br>Advanced integration | Open standards (RDF, JSON-LD) to identify things/entities using the first two principles of linked data | Relations among schemas (dataspace level) | Data anonymisation | Structured queries | Canonical identifiers and entity mappings across sources | Entity enrichment |
| *****<br>Full semantic integration, search, and query | Follows all publishing principles of linked data | Full semantic mappings | Usage control | Schema-agnostic question answering | Knowledge graphs semantically link entities to related entities, data, and streams | Data quality improvement |

Dataspaces to get up and running within a smart environment with a low overhead for administrative setup costs (e.g. catalog, entity management, search and query, and data service discovery). The services follow the 5 star pay-as-you-go model for tiered services.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

# Chapter 6
# Catalog and Entity Management Service for Internet of Things-Based Smart Environments

**Umair ul Hassan, Adegboyega Ojo, and Edward Curry**

**Keywords** Entity management · Knowledge graphs · Data catalog · Dataspace · Internet of Things · Smart environments

## 6.1 Introduction

A fundamental requirement for intelligent decision-making within a smart environment is the availability of information about entities and their schemas across multiple data sources and intelligent systems. This chapter first discusses how this requirement is addressed with the help of catalogs in dataspaces; it then details how entity data can be more effectively managed within a dataspace (and for its users) with the use of an entity management service. Dataspaces provide a data co-existence approach to overcome problems in current data integration systems in a pay-as-you-go manner. The idea is to bootstrap the integration with automated integration, followed by incremental improvement of entity consolidation and related data quality. The catalog and entity management services are core services needed to support the incremental data management approach of dataspaces. We provide an analysis of existing data catalogs that can provide different forms of search, query, and browse functionality over datasets and their descriptions. In order to cover the entity requirements, the catalog service is complemented with an entity management service that is concerned with the management of information about entities.

The chapter is organised as follows. Section 6.2 introduces the important role of entity data. Section 6.3 lists the key requirements and challenges of implementing a catalog and entity service in a dataspace. Section 6.4 examines existing catalogs as described in the literature. Section 6.5 details the implementation of a catalog in the dataspace, with Sect. 6.6 detailing the entity management service. Section 6.7 details the access control service, while Sect. 6.8 describes how a data source joins the dataspace. Finally, Sect. 6.9 summarises the chapter.

© The Author(s) 2020

E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_6

## 6.2 Working with Entity Data

Within a smart environment, analytical and operational activities of intelligent systems revolve around entities of interest. For example, within intelligent energy systems, energy consuming entities (i.e. electrical devices, lights, heating units) are the main entities of interest whereas products and customers are the primary entities for intelligent marketing systems. Typically, in a smart environment, the information about core entities is spread across data silos, including inventory systems and customer relationship systems. Consolidation of this information is known to be among the top priorities of data managers [154]. However, successful integration of information requires overcoming the heterogeneity of data that exists at various levels of detail [155]. Consider the example of a marketing analyst who is preparing a report on a set of company products. For this purpose, the analyst has some data available in a spreadsheet on their local computer that needs to be consolidated with data available in the company's billing system. The first challenge in such consolidation exists at the information representation level due to different data formats and semantics of data models used for describing the products. Once both datasets have been converted to a common format and schema, the analyst will need to perform four actions: (1) discover mapping relationship between attributes of product schemas in the spreadsheet and billing system; (2) determine equivalence relationships among products stored in both data sources; (3) merge the values of mapped attributes for equivalent products to generate a consolidated dataset; and (4) clean the resultant dataset for redundant or conflicting attribute values.

There are several process-oriented methodologies and technical tools available to minimise the manual effort required to achieve the analyst's data integration and data quality workflow. However, a fundamental requirement of integration is the availability of exact information about entities and their schemas across multiple data sources. This chapter first discusses how this requirement is addressed with the help of a catalog in a dataspace, it then details how entity data can be more effectively managed within a dataspace (and for its users) with the use of an entity management service.

## 6.3 Catalog and Entity Service Requirements for Real-time Linked Dataspaces

Driven by the adoption of the Internet of Things (IoT), smart environments are enabling data-driven intelligent systems that are transforming our everyday world, from the digitisation of traditional infrastructure (smart energy, water, and mobility), the revolution of industrial sectors (smart autonomous cyber-physical systems, autonomous vehicles, and Industry 4.0), to changes in how our society operates (smart government and cities). To support the interconnection of intelligent systems in the data ecosystem that surrounds a smart environment, there is a need to enable the sharing of data among intelligent systems.

### 6.3.1 Real-time Linked Dataspaces

A data platform can provide a clear framework to support the sharing of data among a group of intelligent systems within a smart environment [1] (see Chap. 2). In this book, we advocate the use of the dataspace paradigm within the design of data platforms to enable data ecosystems for intelligent systems.

A dataspace is an emerging approach to data management that recognises that in large-scale integration scenarios, involving thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources [2]. Within dataspaces, datasets *co-exist* but are not necessarily fully integrated or homogeneous in their schematics and semantics. Instead, data is integrated on an "*as-needed*" basis with the labour-intensive aspects of data integration postponed until they are required. Dataspaces reduce the initial effort required to set up data integration by relying on automatic matching and mapping generation techniques. This results in a loosely integrated set of data sources. When tighter semantic integration is required, it can be achieved in an incremental "*pay-as-you-go*" fashion by detailed mappings among the required data sources.

We have created the Real-time Linked Dataspace (RLD) (see Chap. 4) as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data, knowledge graphs, and real-time stream and event processing capabilities to support large-scale distributed heterogeneous collection of streams, events, and data sources [4].

### 6.3.2 Requirements

To further support data integration and quality, an RLD must hold information about its participant data sources irrespective of whether they contain primarily static datasets or produce streams of highly dynamic data [19, 156]. Among the primary support services of a dataspace, the catalog service is responsible for managing detailed descriptions of all the data sources that form a dataspace [78]. At a basic level, the descriptions must contain information such as the owner, creation date, type of the data, and semantic information about the data source. At a more detailed level, the catalog must also describe the schema of a data source, the query endpoints, the accuracy of data, access licenses, and privacy requirements. Besides descriptions of individual data sources, the dataspace should also maintain descriptions of relationships between data sources in appropriate forms such as bipartite mappings, dependency graphs, or textual descriptions. The catalog must be able to accommodate a large number of data sources; support varying levels of descriptions about data sources and their relationships, and should make descriptions available in both human and machine-readable formats.

The catalog service plays a crucial role in providing information services for participants in the dataspace, including search, browse, and query services. The catalog should also maintain, wherever possible, a basic entity management service in the form of an inventory of the core entities of interest that includes details on their identifier, type, creation date, core attributes, and associated data source. The catalog can then support simple queries that can be used to answer questions about the presence or absence of an entity in a data source or determine which source contains information on a particular entity. Furthermore, assigning canonical identifiers to entities supports data integration and enrichment as part of stream processing algorithms.

The following primary requirements for a catalog and Entity Management Service (EMS) are needed to support the incremental data management approach of dataspaces:

- **Data Source Registry and Metadata:** The requirement to provide a registry for both static and dynamic data sources as well as their descriptions.
- **Entity Registry and Metadata:** The requirement to provide a registry for entities and their descriptions.
- **Machine-Readable Metadata:** The requirement to store and provide metadata about data sources and entities in machine-readable formats using open standards such as JavaScript Object Notation (JSON) and Resource Description Framework (RDF).
- **HTTP-Based Access:** The requirement to allow HTTP access to data source and entity descriptions.
- **Schema Mappings:** The capability to define mappings between schema elements.
- **Entity Mappings:** The capability to define mappings between entities.
- **Semantic Linkage:** The capability to define semantic relationships and linkages among schema elements and entities.

In addition to the above primary requirements, the following secondary requirements are important for the successful and sustained use of the catalog and an EMS:

- **Search and Browse Interface:** The requirement to provide a user interface over the catalog and EMS, which allows searching and browsing over all the elements stored.
- **Authentication and Authorisation:** The requirement to verify the credentials of users and applications accessing the catalog and EMS which can limit access to sources/entities based on access policies or rules.
- **Data Protection and Licensing:** The requirement to fulfill the privacy and confidentiality requirements of data owners and providing licensing information on the use of data.
- **Keyword Queries**: The requirement to support keyword-based queries over all the data stored in the catalog and EMS.
- **Provenance Tracking:** The requirement of tracking lineage of changes made to the catalog and EMS by users and applications.

## 6.4 Analysis of Existing Data Catalogs

This section provides a short analysis of some existing software and platforms that can be used for implementing data catalogs. The objective of this analysis is to provide a high-level overview of these software packages and their coverage of primary and secondary requirements identified in the previous section. This analysis focuses on a selected list of open-source software while readers are directed towards relevant industry reports to assess proprietary software [157]. In terms of data management, most commercial data catalogs have been developed over existing Master Data Management (MDM) solutions of software vendors. This shift from MDM to data catalogs is primarily driven by the concept of data lakes, which is an industry term used to refer to a loose collection of heterogeneous data assets in enterprises.

Table 6.1 lists the open-source software included in the analysis. QuiltData allows the creation and sharing of data packages using Python. The Comprehensive Knowledge Archive Network (CKAN) is primarily designed for implementing data portals for organisations that publish and share data. CKAN is widely used by public sector organisations and governments to publish open datasets. Dataverse is a web-based platform for data preservation and citation developed at Harvard University. It is primarily used to create a citable reference to a dataset that can be used in publications.

Similarly, the DSpace platform is designed to serve as a repository of digital assets, including multimedia, documents, and datasets. Another software for sharing and preserving research outputs is Zenodo, developed and maintained by CERN. By comparison, the Kylo project form Teradata is designed from a data integration perspective that includes a metadata registry for data sources. The difference between Kylo and other software is evident by the fact that Kylo has been developed by an industry leader where other software primarily originate from academia.

A quick analysis of Table 6.2 reveals that most of the open-source software is limited to addressing the requirements of maintaining a data registry and providing machine-readable access to metadata through HTTP. Most of the catalogs do not address the requirement to manage entity information and the need to provide mappings between schemas and entities. All of the catalogs, except CKAN and Kylo, provide a registry of datasets that are stored internally by the software. On the other hand, both CKAN and Kylo also register external data sources, thus addressing

**Table 6.1** Existing open-source software for data catalogs

| Open-source software | URL | License |
|---|---|---|
| QuiltData | https://github.com/quiltdata/quilt | Apache |
| CKAN | https://github.com/ckan/ckan | AGPL |
| Dataverse | https://github.com/IQSS/dataverse | Apache |
| DSpace | https://github.com/DSpace/DSpace | BSD |
| Zenodo | https://github.com/zenodo/zenodo | GPL |
| Kylo (Teradata) | https://github.com/Teradata/kylo | Apache |

**Table 6.2** Overview of requirements coverage of open-source data catalogs

| | Primary | | | | | | | Secondary | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Requirements | Data source registry and metadata | Entity registry and metadata | Machine-readable metadata | HTTP-based access | Schema mappings | Entity mappings | Semantic linkage | Search and browse interface | Authentication and authorisation | Data protection and licensing | Keyword queries | Provenance tracking |
| QuiltData | + | – | + | ++ | – | – | – | + | ++ | – | ++ | + |
| CKAN | ++ | – | ++ | ++ | – | – | – | ++ | ++ | + | ++ | + |
| Dataverse | + | – | + | ++ | – | – | – | ++ | ++ | + | ++ | + |
| DSpace | + | – | + | ++ | – | – | – | ++ | ++ | + | + | + |
| Zenodo | + | – | + | ++ | – | – | – | ++ | ++ | + | ++ | + |
| Kylo | ++ | – | + | ++ | + | – | – | ++ | + | + | + | + |

++ requirement is well covered
+ requirement is partially covered
– requirement is not covered

a key requirement of a catalog in a dataspace. In terms of machine-readable data, all catalogs provide access to the metadata in JSON format, and CKAN provides data in RDF format.

In terms of secondary requirements, all data catalogs address the requirements with almost full coverage. However, data protection and provenance tracking requirements are only partially addressed by all software. Data protection and licensing requirements are mainly addressed by associating licenses with datasets or data sources in the catalog. Provenance tracking is only limited to the changes made to the metadata instead of the dataset or data source.

## 6.5 Catalog Service

Based on the coverage of the primary and secondary requirements identified, CKAN was chosen as the base software to create the catalog service for the RLD. The catalog extends the CKAN portal with the additional functionally necessary to cover the primary requirements for the RLD. The catalog service provides a registry of the following:

- **Datasets:** A dataset contains contextual information about a building or a thing within a smart environment, real-time sensors data, enterprise data (e.g. customer data, enterprise resource planning systems), or open data such as weather forecast data.
- **Entities:** An entity defines a concrete instance of a concept within the smart environment (e.g. a sensor or a water outlet). The catalog tracks critical entities in the smart environment and links them with the datasets and streams that contain further information about the entities. Metadata about an entity includes the identifier, entity type, and associated datasets.
- **Users and Groups:** Individual users can include data managers, data analysts, and business users who might belong to one or more groups divided along organisational structures or projects.
- **Applications:** Applications are the descriptions of software and services that utilise the dataspace and its services. For example, mobile applications, public displays, data services, analytic tools, web applications, and interactive dashboards.

### *6.5.1 Pay-As-You-Go Service Levels*

Dataspace support services follow a tiered approach to data management that reduces the initial cost and barriers to joining the dataspace. When tighter integration into the dataspace is required, it can be achieved incrementally by following the service tiers defined. The incremental nature of the support services is a core enabler of the pay-as-you-go paradigm in dataspaces. The tiers of service provision provided by the catalog in the RLD follows the 5 star pay-as-you-go model (detailed in Chap. 4). The level of service provided by the catalog increases as follows:

*1 Star* **Registry:** A simple registry of datasets and streams, only pointing to the interfaces available for access.

*2 Stars* **Metadata:** Describing datasets and streams in terms of schema and entities in a non-machine-readable format (e.g. PDF document).

*3 Stars* **Machine-readable:** Machine-readable metadata and simple equivalence mappings between dataset schemas to facilitate queries across the dataspace.

*4 Stars* **Relationships:** Relations among schemas and concepts across the dataspace.

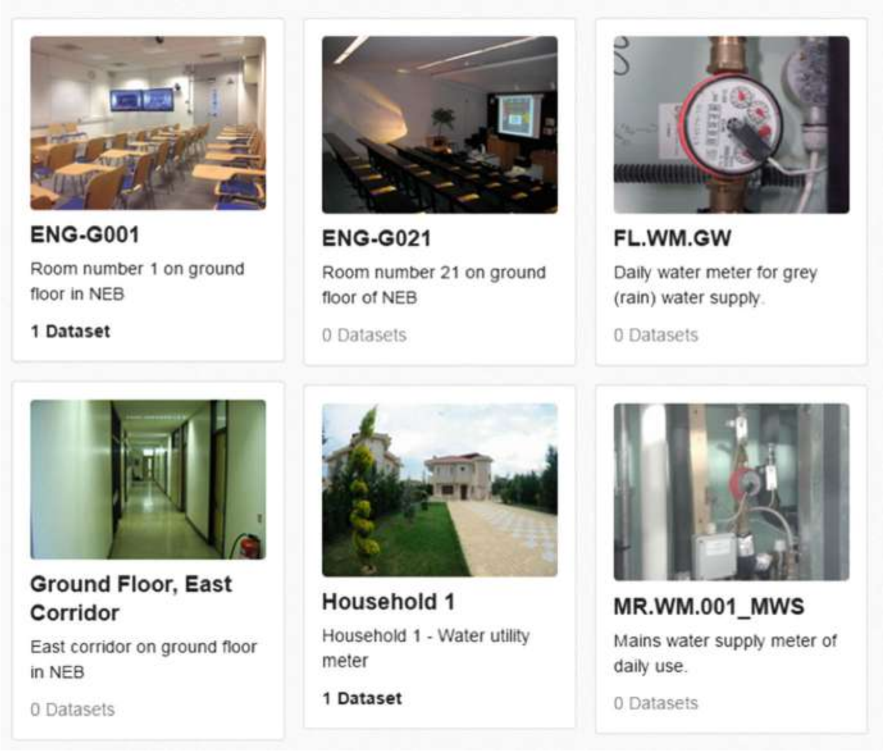
*5 Stars* **Semantic Mapping:** Semantic mappings and relationships among domains of different datasets; thus, supporting reasoning and schema agnostic queries.

The main requirement not covered by CKAN was the need for more advanced support for entity management within the RLD (e.g. entity registry, schema and entity mapping, and semantic linkage). In order to cover these entity requirements, the catalog service in the RLD is complemented with an entity management service that is concerned with the management of information about entities.

## 6.6 Entity Management Service

Managing information about the critical entities in a smart environment is an essential requirement for intelligent decision-making applications that rely on accurate entity information. Similar to MDM, there have been efforts to develop web-scale authoritative sources of information about entities, for example, Freebase [158] and DBpedia [159]. These efforts followed a decentralised model of data creation and management, where the objective was to create a knowledge base. A similar authoritative source of entity information within a dataspace would significantly improve the experience of working with entity data.

Fundamental to the RLD approach is to treat entities as first-class citizens (as illustrated in Fig. 6.1) in the dataspace, which is achieved by using entity-centric knowledge graphs and support from the EMS. The EMS is concerned with the maintenance of information about entities within the smart environment and together with the catalog service acts as the canonical source of entity (meta)data. The EMS facilitates sharing and reusing of entity data within the RLD using (1) a knowledge-graph entity representation framework for structuring entity data and (2) standard ontology languages for defining the semantics of data [160]. Ontologies, also referred to as vocabularies, provide a shared understanding of concepts and entities within a domain of knowledge which supports automated processing for data using reasoning algorithms.



**ENG-G001**

Room number 1 on ground floor in NEB

**1 Dataset**

**ENG-G021**

Room number 21 on ground floor of NEB

0 Datasets

**FL.WM.GW**

Daily water meter for grey (rain) water supply.

0 Datasets

**Ground Floor, East Corridor**

East corridor on ground floor in NEB

0 Datasets

**Household 1**

Household 1 - Water utility meter

**1 Dataset**

**MR.WM.001_MWS**

Mains water supply meter of daily use.

0 Datasets

**Fig. 6.1** Example of managed entities in the Entity Management Service [4]

The relationship of entities across data sources and intelligent systems in a smart environment can quickly become complicated due to the barriers of sharing knowledge among intelligent systems. This is a significant challenge within traditional data integration approaches and the use of linked data and knowledge graph techniques that leverage open protocols and W3C that can support the crossing of knowledge boundaries when sharing data among intelligent systems.

The EMS leverages the principles of linked data from Tim Berners-Lee (see Chap. 2) [41] and adapts them to the management of entities. Thus, the EMS has the following “Linked Entity” principles:

- **Naming:** Each managed entity within the EMS is identified using a Uniform Resource Identifier (URI). Managed entities can be a person, a building, a device, an organisation, an event or even concepts such as risk exposure or energy and water consumption.
- **Access:** Each managed entity within the EMS can be accessed via an HTTP-based URI which can be used to retrieve detailed entity data.
- **Format:** When an entity URI is looked up (i.e. dereferenced) to retrieve entity data, useful information about the entity is provided using open-standard formats such as RDF or JSON-LD.

- **Contextualisation:** Entity data includes URIs to other entities so that more information can be discovered on-the-fly. Referencing other entities, through URIs, thus creates a knowledge graph that could be traversed by automated software to discover and link information with the dataspace.

### 6.6.1 Pay-As-You-Go Service Levels

Similar to the tiered approach used by the catalog, the level of active entity management follows the 5 star pay-as-you-go model of the RLD. The entity management service has the following levels of incremental support:

*1 Star* **No Service:** Entities are not managed.

*2 Stars* **Documented:** Entity descriptions (e.g. schema and identifiers) are documented in a non-machine-readable format (e.g. PDF document).

*3 Stars* **Source-level:** Machine-readable entity at the source-level.

*4 Stars* **Multi-source Mapping:** Canonical identifiers for entities in the dataspace and mapping across sources.

*5 Stars* **Entity Knowledge Graphs:** Entities are semantically linked to other related entities, data, and streams across the dataspace to create a knowledge graph.

### 6.6.2 Entity Example

The EMS follows the incremental dataspace philosophy; in practice, you only connect data sources related to an entity on an as-needed basis. The approach encourages that entities should be as minimal as possible to achieve the desired results. Figure 6.2 describes a minimal data model for entities in one of the smart water pilots.

The key entities of the data model and the sources they originate from are:

- **Sensor:** Measures the flow of water and generates a stream of data to calculate the water consumption levels of the area covered by the sensor (from the Internet of Things Platform).
- **Observation:** The sensor output including the units and rate of measurement (from the Internet of Things Platform).
- **Outlet:** Information on the actual physical water outlet is necessary for analysis and decision-making. It is possible that a single sensor might be installed for a set of outlets. In such cases, a cumulative assessment of water consumption is needed (outlet description crowdsourced using human task service).
- **Location:** Information on the associated spatial locations serviced by the water pipe (from Building Management System).
- **User Group:** Each sensor is associated with a set of users who have permission to access the data (from Enterprise Access Control). The access control service leverages this information.

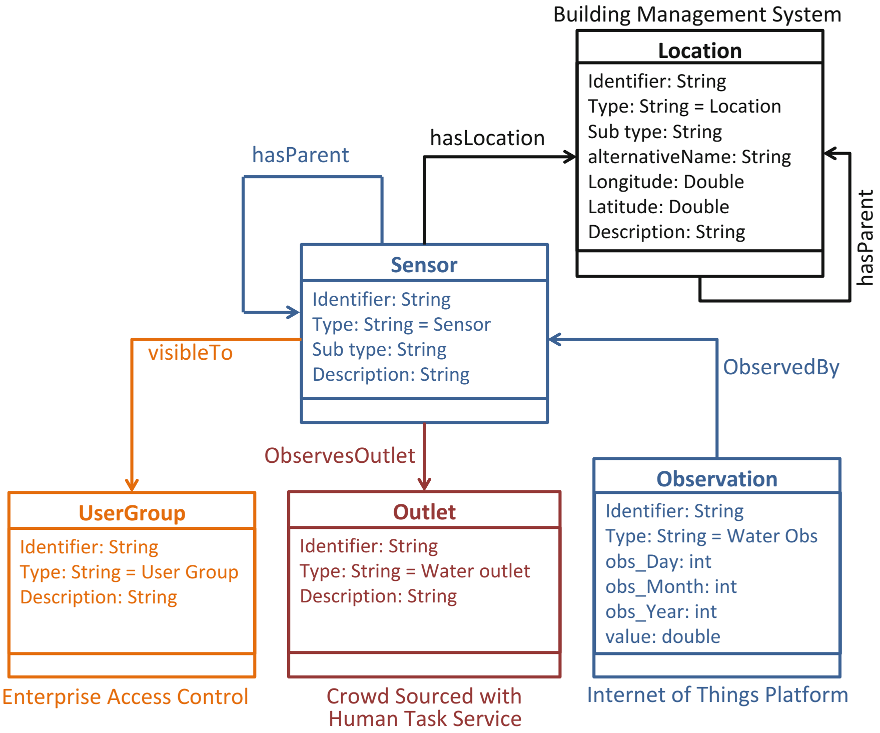


**Fig. 6.2** Minimal data model for entities in an intelligent water management system [4]

## 6.7 Access Control Service

The access control service ensures secure access to the data sources defined in the catalog. Access is managed by defining access roles for applications/users to the data source/entity that are declared in the catalog. The access control service is an intermediary between the applications/users and the dataspace by using the catalog as a reference to verify access for applications/users to the actual data sources. The advantage of this approach is to keep the applications/user's profiles centrally managed by the catalog under the governance of the dataspace managers. Within the pilot deployments, we defined three types of roles for access control: (1) dataspace managers, (2) application developers/data scientists, and (3) end-users. To simplify the process of securely querying data sources, the access control service offers a secure query service to applications. As illustrated in Fig. 6.3, the workflow of an application using the secure query capability of the access control service and the roles of the users are as follows[1]:

[1] Demo Video available at this link: https://www.youtube.com/watch?v=KukUd5VCheY

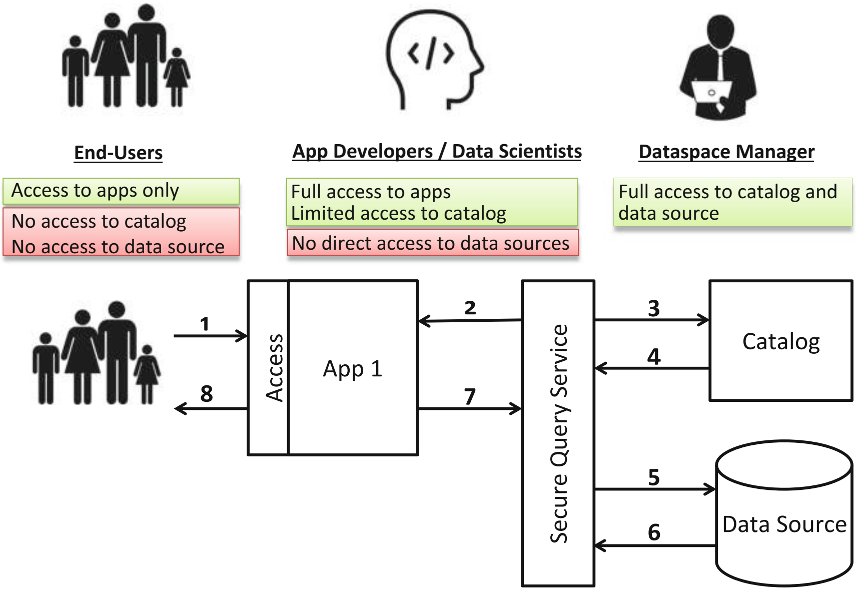


**Fig. 6.3** Query workflow using the access control service [4]

1. The user connects to the application (App1).
2. The application maps the user ID to its profile and access as the secure query service via an identification token (API Key).
3. The query service verifies the application ID and its API Key and checks that it has the right to access the data source (e.g. dataset or an entity).
4. Authorisation results are sent back to the query service.
5. If the user is authorised, the query service gets the data from the source.
6. Results from the data source are sent back to the query service.
7. The query service sends the data to the application.
8. The application returns the data to the user (e.g. via a UI or a file).

### *6.7.1 Pay-As-You-Go Service Levels*

In terms of tiered levels of support for the access control service, this is defined by the capability to increasingly limit access to more fine-grained levels within a data source. The access control service has the following levels of service:

*1 Star* **No Service:** The access control service does not manage the source.
*2 Stars* **Coarse-grained:** Access is limited to the user at the dataset level.
*3 Stars* **Fine-grained:** Access is limited to users at the entity level with the use of the secure query service.

*4 Stars* **Data Anonymisation:** Access to sanitised data for privacy protection. (Not supported in pilots).

*5 Stars* **Usage Control:** Usage of the data is controlled as it moves around the dataspace. (Note: This functionality is not currently implemented).

## 6.8 Joining the Real-time Linked Dataspace

The RLD is composed of multiple data sources, including real-time sensor streams, historical databases, large text files, and spreadsheets. The RLD adopts a pattern in which the publisher of the data is responsible for paying the cost of joining the dataspace. This is a pragmatic decision as it allows the dataspace to grow and enhance gradually. For a data source to become a part of the dataspace, it must be discoverable and must conform to at least the first star rating of the RLD. The registration process entails detailing the metadata of the source, which helps users of the catalog in locating and using the data source. A seven-step approach has been defined for including a data source into the RLD (see Fig. 6.4). The seven steps are Register, Extract/Access, Transform, Load, Enrich, Map, and Monitor (RETLEMM). Some of the steps are optional and depend on the capability of the data source to meet requirements around machine-readable data and query & search capabilities and interfaces. The RETLEMM steps are:

- **Register:** A new data source joining the dataspace would require it to be registered in the dataspace catalog. The registration means that the catalog contains an entry describing the data source at a minimum regarding type, access, and format. Completion of this step will give the data source a rating of one star and the data source is considered part of the dataspace since it can be accessed and

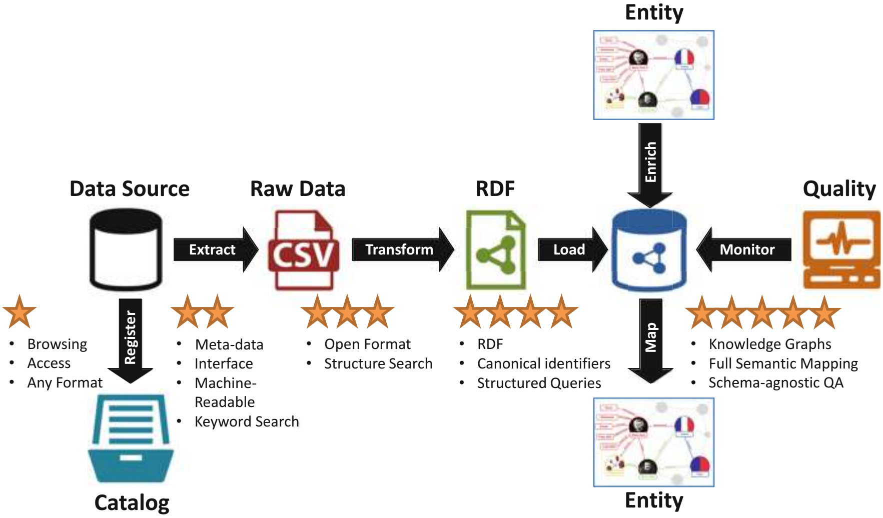


**Fig. 6.4** RETLEMM process for a data source to join the RLD

used. Further optional information about the data source can include the physical address of files, a query interface/endpoint, additional metadata, and entity data.
- **Extract/Access:** The second step of the process is to allow access to the data from the data source in a machine-readable format; this will rate the source as a minimum of 2 stars. How data is accessed depends on the data source; for simple sources with limited capability (e.g. Excel), the data may need to be extracted. For more sophisticated data sources (e.g. database), the data may be accessible via a query interface. To demonstrate all the process steps in this example (see Fig. 6.4), it is assumed that information is extracted in the form of CSV files. The use of an open format will move it to 3 stars.
- **Transform:** Given the CSV representation, the next step is to convert the data into an appropriate format for publishing. A simple semi-automated process for transforming the CSV files to RDF files is possible using tools such as Microsoft Excel and OpenRefine. A similar process can be used to perform an on-the-fly transformation of the results of a database query (see Adapters below). This step moves the data towards 4 stars.
- **Load:** Once the data has been converted and represented in the RDF format, the next step is to store it in an appropriate data store. For this step, any general-purpose RDF store may be used. However, it is necessary for the RDF store to have the necessary publishing, querying, and search functionalities to support applications. This step is not necessary where the data source has a queryable interface and results can be transformed into on-the-fly RDF. The data is now 4 stars.
- **Enrich:** The above steps are enough to support analytical and decision support applications. Nevertheless, it is desirable to enhance the metadata with additional information such as links to related entities in other datasets. This optional step adds contextual information to achieve the overall entity-centric vision of the RLD. The data will move towards 5 stars.
- **Map:** Similar to the enrich step, the schema and entities of a data source may be mapped to other data sources and entities in the catalog. This facilitates integration and deduplication of classes and entities. Also, it allows the automated processing of data collected from multiple datasets using advanced reasoning and schema agnostic query tools. The data will now be 5 stars.
- **Monitoring:** It is not unusual for a data source to change or update its definitions and attributes. These changes can introduce data quality issues and errors which can affect the performance of the dataspace. The RLD utilises a simple monitoring process to check for changes in data sources in terms of availability and data quality.

When a data source joins the dataspace, the RETLEMM process can be performed manually by the data source owners with the help of the dataspace support services. However, the Extract, Transform, and Load (ETL) steps can be automated to speed up the process. Automation is desirable for large-scale historical data and real-time metering data. In the following, we discuss the two alternatives for automation:

- **Adapters:** Adapters can be considered a non-materialised view of a data source. They encode the ETL process in the form of mappings between the source data format and the target data format. In the case of a historical database, the data resides in the source, and the ETL is performed on-the-fly every time queries are posted on a non-materialised view. In the case of a real-time data stream, the ETL is performed on-the-fly as data is generated by the streaming source.

- **Scheduled Jobs:** This form of the ETL process is performed either once for a large static database or periodically for a large dynamic database. It is a common activity among existing data warehouse implementations.

## 6.9 Summary

This chapter underlines the need for a catalog service for successful implementation of Real-time Linked Dataspaces. Specifically, it is established that the catalog should not only serve as a registry of data sources in a dataspace but also provide an entity management service. Based on a set of requirements identified for the catalog, a short analysis of existing open-source software is provided to assess their coverage of requirements. The design of the catalog and entity management service for the Real-time Linked Dataspace is detailed including aspects such as tiered services levels, entity modelling, access control, and the process for a data source to join the dataspace using the catalog and entity management service.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

# Chapter 7
# Querying and Searching Heterogeneous Knowledge Graphs in Real-time Linked Dataspaces

André Freitas, Seán O'Riáin, and Edward Curry

**Keywords** Knowledge graphs · Query processing · Data search · Best-effort · Dataspace

## 7.1 Introduction

As the volume and variety of data sources within a dataspace grow, it becomes a semantically heterogeneous and distributed environment; this presents a significant challenge to querying the dataspace. Approaches used for querying siloed databases fail within large dataspaces because users do not have an a priori understanding of all the available datasets. This chapter investigates the main challenges in constructing query and search services for knowledge graphs within a linked dataspace. Search and query services within a linked dataspace do not follow a one-size-fits-all approach and utilise a range of different techniques to support different characteristics of data sources and user needs.

This chapter is structured as follows: Section 7.2 explores the difference between querying and searching knowledge graphs in a Real-time Linked Dataspace and details the high-level functionality needed by the search and query service. Section 7.3 introduces search and query over dataspaces, discusses the challenges with data heterogeneity in a dataspace, and identifies the core requirement for the search and query service. State-of-the-art analysis of existing approaches to searching and querying is provided in Sect. 7.4. Section 7.5 details an analysis of the emerging design features for creating schema-agnostics query mechanisms, and the chapter concludes in Sect. 7.6.

© The Author(s) 2020

E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_7

## 7.2 Querying and Searching in Real-time Linked Dataspaces

Driven by the adoption of the Internet of Things (IoT), smart environments are enabling data-driven intelligent systems that are transforming our everyday world, from the digitisation of traditional infrastructure (smart energy, water and mobility), the revolution of industrial sectors (smart autonomous cyber-physical systems, autonomous vehicles, and Industry 4.0), to changes in how our society operates (smart government and cities). To support the interconnection of intelligent systems in the data ecosystem that surrounds a smart environment, there is a need to enable the sharing of data among intelligent systems.

### *7.2.1 Real-time Linked Dataspaces*

A data platform can provide a clear framework to support the sharing of data among a group of intelligent systems within a smart environment [1] (see Chap. 2). In this book, we advocate the use of the dataspace paradigm within the design of data platforms to enable data ecosystems for intelligent systems.

A dataspace is an emerging approach to data management that is distinct from current approaches. The dataspace approach recognises that in large-scale integration scenarios, involving thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources [2]. Within dataspaces, datasets *co-exist* but are not necessarily fully integrated or homogeneous in their schematics and semantics. Instead, data is integrated on an *as-needed* basis with the labour-intensive aspects of data integration postponed until they are required. Dataspaces reduce the initial effort required to set up data integration by relying on automatic matching and mapping generation techniques. This results in a loosely integrated set of data sources. When tighter semantic integration is required, it can be achieved in an incremental *pay-as-you-go* fashion by detailed mappings among the required data sources.

We have created the Real-time Linked Dataspace (RLD) (see Chap. 4) as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data and real-time stream and event processing capabilities to support a large-scale distributed heterogeneous collection of streams, events, and data sources [4]. In this chapter, we focus on the search and query support services of the RLD.

Dataspaces assume that the querying capability of the participants in the dataspace is not equal, and they do not assume the support of any specific standards to support data sharing. By building on web (URIs and HTTP) and semantic web standards (such as the Resource Description Framework and RDF Schema [RDFS]), and vocabularies, RLD can effectively reduce barriers to data publication, consumption, and reuse within a dataspace. Participants in a linked dataspace expose their data as Knowledge Graphs (KGs), which can be interlinked and integrated with other datasets, creating an interlinked dataspace.

RLDs and generic dataspaces share more commonalities than differences, and the analysis provided in this chapter is relevant to both. In this chapter, the scope of the search and query service is mainly focused on non-streaming data sources with the search and query of live data streams discussed in Part III of this book. The discussion in this chapter builds on our earlier analysis of querying heterogeneous linked data sources [111] by contextualising the challenges within Real-time Linked Dataspaces [4].

### *7.2.2 Knowledge Graphs*

Knowledge graphs pose challenges inherent to querying highly heterogeneous and distributed data. To query, data users must first be aware of which datasets potentially contain the data they want and what data model describes these datasets, before using this information to create structured queries. This query paradigm is deeply attached to the traditional perspective of structured queries over databases and does not suit the heterogeneity, distributiveness, or scale we expect from the datasets and KGs within a linked dataspace. It is impractical to expect users to have a previous understanding of the structure and location of datasets within the linked dataspace. Letting users expressively query relationships in the data while abstracting them from the underlying data model is a fundamental problem for massive data consumption, which, if not addressed, will limit the utility of dataspaces for consumers.

Consider a journalist compiling a list of facts regarding public personalities and their family connections. The journalist can express his or her information needs as natural language queries, such as “Who are the children of Marie Curie married to?” Document search engines cannot currently provide a level of query interpretation that could point directly to the final answer. With a traditional search engine, the journalist must navigate through the links and read the content of each candidate page the search engine returns.

The information that can answer this query may be available in the linked dataspace. However, to access it, users must know the location and structure of relevant datasets and the syntax of the query language. There exists a semantic gap between the user’s information need, which is expressed in a generic natural language query and the data representation in the target dataset. The query’s terms and structure differ from the data representation in the dataset. The provision of intuitive and flexible query mechanisms that can approximate users from an unconstrained amount of data represents a fundamental challenge of querying knowledge graphs in a linked dataspace.

### *7.2.3 Searching Versus Querying*

Query mechanism for structured data which supports data consumers with expressive queries (queries which can make use of the conceptual structure behind the database and the supported database operations) and abstracts them away from the representation (being schema-agnostic) is an active research challenge.

The simplicity and intuitiveness of search engine interfaces, where users search the web using keyword queries, was a crucial element in the widespread adoption of web search engines and in the process of maximising the value of the information available on the web. On the other side of the spectrum, from the perspective of structured/semi-structured data consumption, users expect precise and expressive queries. In this scenario, most users query data with the help of structured query languages such as SQL or SPARQL. In a large-scale data scenario, structured query approaches do not thoroughly address all search and query usability requirements from all categories of users (such as being accessible to casual users and supporting lower query construction times for expert users).

With the web, users have recognised search to be a first-class activity. The search paradigm used in the web, however, cannot be directly transported for querying structured data. Keyword search over data does not provide the desired expressivity, while traditional structured query mechanisms have poor usability. Query expressivity and usability are two dimensions of database querying which define trade-off behaviour. Different categories of query/search approaches have emerged, targeting the trade-off between usability and expressivity (see Fig. 7.1) and have achieved some level of success.

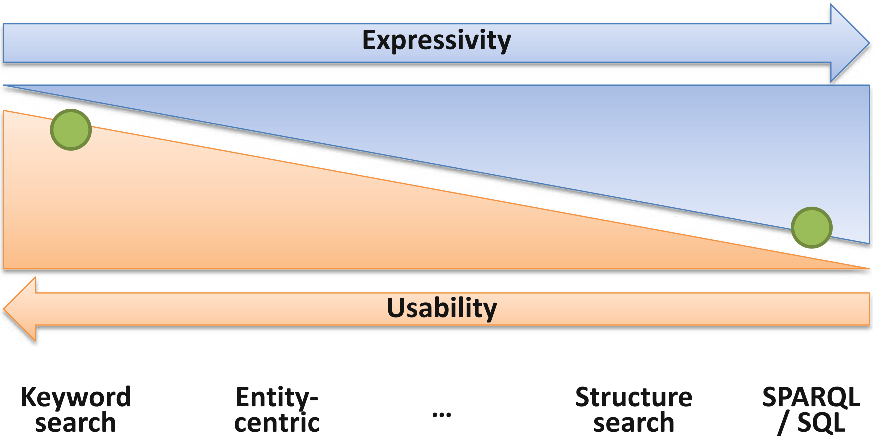


**Fig. 7.1** The expressivity–usability trade-off for querying over structured data. The green dots indicate that an ideal query mechanism must provide both high expressivity and high usability [111]. Adapted from [161]

### 7.2.4 Search and Query Service Pay-As-You-Go Service Levels

The objective of the Search and Query service is to help developers, data scientists, and users to find relevant datasets within the dataspace. Users can navigate the dataspace by entities (if supported), or by performing a search or query on the datasets. A key challenge in developing search and query services over heterogeneous sources in a dataspace is the expressivity–usability trade-off. An ideal dataspace query mechanism must provide both high expressivity and high usability. As data sources are more tightly integrated into the dataspace, and move towards forming a knowledge graph, the search and query service can offer more sophisticated functionality.

Dataspace support services follow a tiered approach to data management that reduces the initial cost and barriers to joining the dataspace. When tighter integration into the dataspace is required, it can be achieved incrementally by following the service tiers defined. The incremental nature of the support services is a core enabler of the pay-as-you-go paradigm in dataspaces. The functionality of the search and query service follows the five star pay-as-you-go model (detailed in Chap. 4) of the RLD. The search and query service offers the following levels of functionality:

*1 Star* **Browsing:** Browsing of the datasets available in the dataspace catalog.

*2 Stars* **Keyword Search:** Basic keyword search of the sources within the dataspace.

*3 Stars* **Structure Search:** A structured search is when the dataset has been indexed by the search service to enable entity-centric searches over the data and structure of the dataset.

*4 Stars* **Structured Queries:** Structured queries are possible where the data source supports a SPARQL interface, or the data source has been loaded into the local RDF store of the query service. In order to write a structured query (which can be entity-centric), the user must understand the underlying schema of the data.

*5 Stars* **Schema-Agnostic Question Answering:** A best-effort entity-centric natural language interface to the dataspaces knowledge graph that allows users to ask questions without understanding the underlying schema.

The provision of search and query services within a linked dataspace does not follow a one-size-fits-all approach with a range of different techniques used to support the different characteristics of the data sources and user needs. Running queries over heterogeneous data sources is a particularly challenging proposition that is an active area of research. The two initial levels, Browsing and Keyword search, are well-understood techniques that have readily available solutions. The third level is structured queries where keyword queries' expressivity is enhanced to include the structure of the data. This is achieved by extending existing inverted list indexes to represent structural information present in datasets. The next level is structured queries where SPARQL query support is provided over data sources via endpoints on the data source or by importing data into the local RDF store. These approaches are suited to creating queries over homogenous and well-formed schemas that the user understands. Working with heterogeneous sources at large scales requires a different approach with the ability for the query mechanism to be agnostic to the underlying diverse schemas. At the high-end of the search and querying service for the RLD is schema-agnostic question answering over the interlinked knowledge graph that simplifies user–data interaction. A famous example of this emerging style of data interaction is the IBM Watson Question Answering (QA) system which competed in the television game show Jeopardy or the Apple Siri virtual assistant.

## 7.3 Search and Query over Heterogeneous Data

The vocabulary problem for databases is a consequence of data heterogeneity [162], that is, the multiple realisations in which data can be represented. Even if given the same task, different database designers can materialise the same domain into a database using different lexical expressions, conceptualisations, data models, data formats, or record granularities [162]. This intrinsic variability in the construction of a database defines a fundamental level of data heterogeneity between different databases.

Similarly, there is an intrinsic heterogeneity between a specific data representation and the data consumer's mental representation of a domain. If asked to materialise their information needs as free queries (e.g. using natural language), data consumers would be likely to use different terms and structures in the query formulation, a fact which is supported by Furnas et al. [163]. The intrinsic heterogeneity is mediated by the role of phenomena intrinsic to natural languages such as synonymy, ambiguity, and vagueness. The vocabulary problem is a concrete instance of the syntactic and semantic barriers in the knowledge boundaries identified in the Knowledge Value Ecosystem (KVE) Framework that exist when sharing knowledge among systems. These boundaries to knowledge sharing are discussed in more detail in Chap. 2.

### *7.3.1 Data Heterogeneity*

Data heterogeneity becomes a more immediate concern as users start to query data/KGs from different datasets built by independent parties. This is the scenario faced by dataspaces (and Knowledge Graphs) where one starts to move from a centralised schema and data model (where data is integrated under a single representation model) to a decentralised scenario where data from different schemas and data models are brought together into a different data consumption context [2, 74]. The concept of data heterogeneity can be examined within different dimensions:

- *Conceptual Model Heterogeneity:* Different domains can be conceptualised using different abstractions and lexical expressions, which are dependent on the intended use behind the database and the background of the individuals modelling the domain (the KVE knowledge boundary). Given a modelling task with a minimum level of complexity, it is unlikely that two independent parties will generate identical conceptual models [162, 163]. Semantic heterogeneity emerges as a central concern in a dataspace when, data from multiple datasets, developed by different third-parties, need to be accessed and processed in a different context. Conceptual model heterogeneity includes distinct classes of differences which define the conceptual gap.
- *Format Heterogeneity:* Covers different formatting assumptions for values. This dimension covers the notational and measurement units' differences. Examples of value types dependent on data format are currency, numerical values, and date-time values. Abbreviations and acronyms are also included in this category.
- *Data Model Heterogeneity:* Data models provide the syntactical model in which different data objects are represented. Different data sources can be represented using different data models (the KVE knowledge boundary). Examples of data models include the relational model RDF and eXtensible Markup Language (XML), among others.

The three data heterogeneity dimensions are orthogonal and impact the reconciliation of model dimensions between different databases/KGs and the ability of users to query a data source. The more significant the gap between the two models (data, format or conceptual), the larger is the cost of querying or data integration.

The abstraction of users from the conceptual database model is intrinsically connected with the provision of a principled semantic matching mechanism to cross the conceptual gap between the user query and the data representation. Query mechanisms with the ability to automatically bridge the gap between the user and database conceptual models are described as schema-agnostic or vocabulary-independent queries. A motivational scenario example is introduced below.

### *7.3.2 Motivational Scenario*

Suppose a user has an information need expressed as the natural language query 'Who are the children of Marie Curie married to?' (Fig. 7.2). The person has access to different databases/KGs within a dataspace which contain data that can help to address the information need. However, the data representations inside the target databases do not match the vocabulary and structure of the natural language query.

Figure 7.2 depicts an example of the semantic gap between the example user query and possible representations for the knowledge graphs supporting answers for the query. In (a), 'child' and 'married to' in the query map to 'Child' and 'Spouse' in the knowledge graph; in (b), these query terms map to 'motherOf' and 'wifeOf'

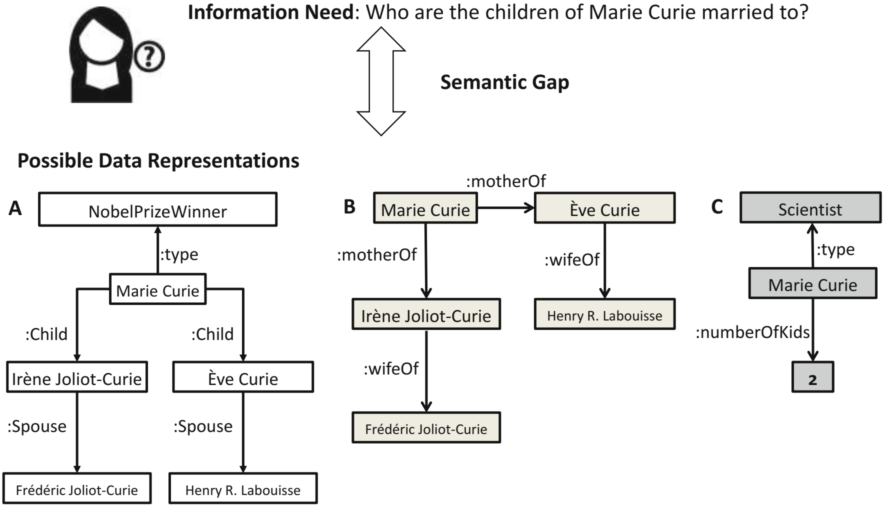


**Fig. 7.2** Example of user information requirement expressed as a natural language query and possible knowledge graph representations in different conceptual models

respectively; while in (c), the query information related to ‘child’ is given by the predicate ’numberOfKids’ representing an aggregation in (c), not fully mapped to the query information need.

To address query-data alignments, it is necessary to provide a query mechanism which can support a semantic matching which copes with the semantic gap between the user query and the knowledge graph representation.

### 7.3.3 Core Requirements for Search and Query

The dimensions of semantic heterogeneity are at the centre of the search and query challenge within dataspaces and addressing them directly can define the semantic matching requirements to provide robust search and query mechanisms. However, in addition to the requirements related to the semantic matching, search and query approaches need to satisfy requirements common to all search and query mechanisms. These requirements are used as qualitative dimensions to evaluate the effectiveness of search and query approaches:

- *High Usability and Low Query Construction Time:* Support for a simple and intuitive interface for experts and casual users.
- *High Expressivity:* Queries referencing structural elements and constraints in the dataset (relationships, paths) should be supported, as well as operations over the data (e.g. aggregations, conditions).

- *Accurate and Comprehensive Semantic Matching:* Ability to provide a principled semantic matching addressing all the dimensions of the semantic heterogeneity problem (abstraction, conceptual, compositional, functional) with high precision and recall.
- *Low Setup and Maintenance Effort:* Easily transportable across datasets/knowledge graphs without significant manual adaptation effort. The query mechanism should be able to work under an open domain and across multiple domains. Databases should be indexed with a minimum level of manual adaptations in the construction of supporting semantic resources used in the semantic matching.
- *Interactive Search and Low Query-Execution Time:* Minimisation of user interaction/feedback effort in the query process. Users should get answers with interactive response times for most of the queries. An interactive query execution time is contrasted with a batch query execution time (seconds vs. minutes).
- *High Scalability:* The query approach should scale to large datasets/knowledge graphs both in query execution and indexing construction time.

With a clear understanding of the challenges and requirements that need to be overcome, we now examine state-of-the-art approaches for searching and querying heterogeneous data.

## 7.4 State-of-the-Art Analysis

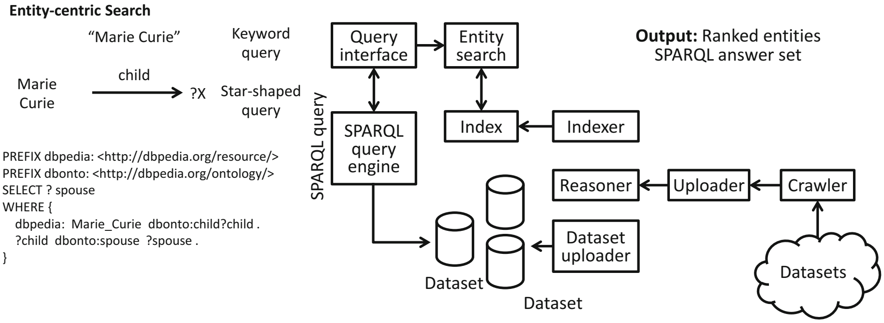
Three high-level categories of approaches for querying heterogeneous data within dataspaces exist: (1) approaches employing strategies inherited from the Information Retrieval (IR) space in which keyword search is mixed with elements from structure queries, (2) approaches focusing on natural language queries, and (3) structured SPARQL queries over distributed datasets. Leveraging existing work [111] we focus on the usability and semantic matching problems, thus analysing approaches from the first two categories.

### *7.4.1 Information Retrieval Approaches*

We can categorise IR approaches according to the index type, which includes entity-centric search approaches and structure search approaches. Although both types provide hybrid search interfaces that merge keyword search with dataset structure elements, only structure search targets indexing strategies focus on addressing the expressivity–usability trade-off at the index construction level.

**Entity-Centric Search**

Entity-centric approaches let users search for entities (instances and classes) in datasets, employing Vector Space Model (VSM) variations to index those entities. Existing approaches range from less expressive queries, based on keyword search



```
PREFIX dbpedia: <http://dbpedia.org/resource/>
PREFIX dbonto: <http://dbpedia.org/ontology/>
SELECT ? spouse
WHERE {
   dbpedia:  Marie_Curie  dbonto:child?child .
   ?child  dbonto:spouse  ?spouse .
}
```

**Fig. 7.3** High-level architecture components for Sindice (entity-centric search) [111]

over textual information associated with the dataset entities, to star-shaped queries and hybrid queries (i.e. queries mixing keyword search, and structured queries centred on an entity).

The Semantic Web Search Engine (SWSE) is a search and query service that implements an architecture with components for crawling, integrating, indexing, querying, and navigating over multiple data sources [164]. The system architecture's main components include query processing, ranking, an index manager, and an internal data store (YARS2), which focuses on scalability issues to enable federated queries over linked data. SWSE uses an approach called ReConRank to rank entities [164]; this approach adapts the PageRank algorithm to work over RDF datasets, propagating dataset-level scores—computed from interlinking patterns—to data-level entities. The Scalable Authoritative OWL Reasoner (SAOR) provides an RDFS and a partial Web Ontology Language (OWL) reasoning engine to address scalability issues [164]. SAOR applies reasoning only on dataset fragments supported by an authoritative ontological definition.

Sindice is a search and query service for the linked data web that ranks entities according to the incidence of keywords associated with them [165]. It uses a node-labelled tree model to represent the relationship among datasets, entities, attributes, and values. Similar to SWSE, Sindice provides a comprehensive entity-centric search and indexing approach. Figure 7.3 depicts Sindice's architecture.

The SPARK [166] approach provides a ranking solution for translating keyword-based queries to low complexity SPARQL queries, targeting low complexity RDF datasets. The SPARK is based on three basic steps: term mapping, query graph construction, and query ranking.

Entity-centric search approaches have developed comprehensive data management strategies for linked data on the web, providing the infrastructure for managing the complete crawl–index–search cycle. These approaches also developed services complementary to the entity-centric search process that let users either visually explore (via Visinav [164] and Sigma [165]) or execute full structured SPARQL queries over the crawled data. Entity-centric approaches avoid significant changes in

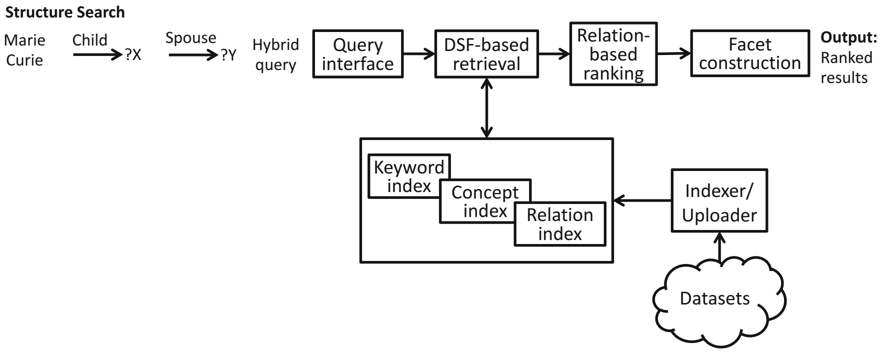


**Fig. 7.4** High-level architecture components for Semplore (structure search) [111]

standard indexing strategies, inheriting index and search optimisation mechanisms present in existing VSM frameworks. These approaches have avoided tackling the expressivity–usability trade-off by aggregating multiple query interfaces; in practice, to execute expressive queries, users must be aware of the vocabularies behind the datasets. Also, most entity-centric approaches have only limited evaluation in terms of the search result quality.

### Structure Search

Structure search engines improve keyword queries' expressivity, extending existing inverted list indexes to represent structural information present in datasets. The main difference between entity-centric search and structure search is that the latter improves query expressivity with support from the extended index.

The search engine Semplore [167] uses a hybrid query formalism that combines a keyword search with structured queries (i.e. a subset of SPARQL). Semplore uses position-based indexing to index relations and join triples. It relies on three types of inverted indexes: keyword, concept, and relation. Semplore also explores user feedback strategies for improving search, providing a faceted and navigational interface. Figure 7.4 depicts Semplore's high-level architecture. Xin Dong and Alon Halevy propose an approach for indexing triples to enable queries that combine keywords and dataset structure elements [168]. To provide a more flexible semantic matching, the authors propose four structured index types based on the introduction of additional structural information and semantic enrichment in the inverted lists. Taxonomies associated with the dataset vocabularies are used as a semantic enrichment strategy.

Structure search approaches target the expressivity–usability trade-off by modifying and extending traditional inverted index structures. They introduce a limited level of semantic matching by considering the terminology-level information present in datasets or by enriching the index with related terms using WordNet. No comprehensive evaluation of the search results' quality exists, making it unclear how these approaches perform in addressing the expressivity–usability trade-off.

## 7.4.2 Natural Language Approaches

Approaches in the literature based on natural language queries target query mechanisms with high usability and expressivity. Although some approaches focus on the question answering (QA) problem, in which, similar to databases, precise answers are expected as the output. Others focus on a best-effort scenario that returns a ranked list of results.

**Question Answering**

The investigation of QA systems focuses on the problem of allowing users to query data using natural language queries. As opposed to IR techniques' best-effort nature, QA systems target crisp answers, as with structured queries over databases. Work on QA approaches investigates the interpretation of users' information requirement that is expressed as natural language queries, applying Natural Language Processing (NLP) techniques to parse queries and match them with dataset structures. Substantial research efforts have focused on this problem. We look at two works on open domain linked data.

PowerAqua is a QA system that uses PowerMap, a hybrid matching algorithm comprising terminology-level and structural schema-matching techniques with the assistance of large-scale ontological or lexical resources [169]. In addition to the ontology structure, PowerMap uses WordNet-based similarity approaches as a semantic approximation strategy.

Exploring user interaction techniques, FREyA is a QA system that employs feedback and clarification dialogs to resolve ambiguities and improve the domain lexicon with users' help [170]. Compared to PowerAqua, FREyA delegates a large part of the semantic matching and disambiguation process to users. User feedback enriches the semantic matching process by allowing manual entries of query-vocabulary mappings. Figure 7.5 depicts FREyA's high-level architecture.

TBSL [172] exploits both natural language and information retrieval techniques and explores corpus-based patterns to support schema-agnosticism. TBSL relies on parsing the user question to produce a query template. The core rationale behind the approach is that the linguistic structure of a question together with well-defined expressions in the context of QA over structured data (such as more than and the

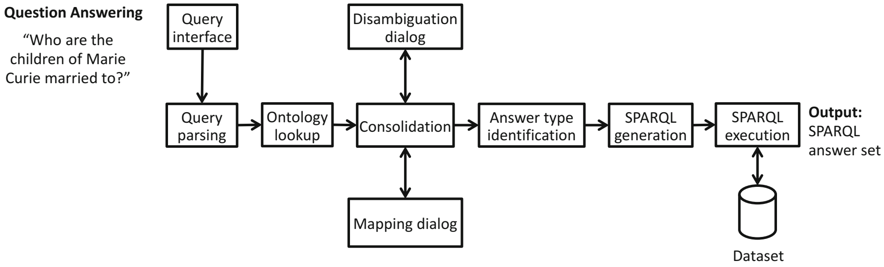


**Fig. 7.5** High-level architecture components for FREyA (question answering) [111]

most) define a domain-independent structure for the query, which then needs to be filled in with domain-specific vocabulary elements.

Compared to IR-based approaches, QA approaches aim toward more sophisticated semantic matching techniques because they target queries with high expressivity and do not assume users are aware of the dataset representations (high usability). In contrast to entity-centric and structure search approaches, QA systems have a strong tradition of evaluating the quality of results and have concentrated less on performance and scalability issues. Traditionally, QA approaches have focused on limited semantic matching (WordNet-based) strategies, making them unable to cope with high levels of heterogeneity. Most QA approaches apply limited semantic matching techniques (e.g. synonymic, taxonomic similarity) for matching query terms to dataset terms. Also, they depend on resources that are manually created (WordNet) and difficult to expand across different domains.

### Best-Effort Natural Language Interfaces

More recent approaches aim to merge natural language queries' expressivity and usability with IR models' scalability and best-effort nature, targeting a best-effort natural language search mechanism. As in QA systems, users can still enter full natural language queries; however, instead of targeting crisp answers, these approaches return an approximate ranked list of results.

The Treo natural language query mechanism for linked data uses semantic relatedness measures derived from Wikipedia to match query terms to dataset terms [171]. The use of semantic relatedness measures allows the quantification of the semantic proximity between two terms, using semantic information which is embedded in large textual resources available on the web such as Wikipedia. Wikipedia-based semantic relatedness measures address previous limitations of WordNet-based semantic matching. Treo's approach combines entity search, spreading activation search, and semantic relatedness to navigate over the linked data/knowledge graph, semantically matching the parsed user query to the data representation in the datasets. Figure 7.6 depicts Treo's components.

The principles of the Treo approach are generalised by constructing a distributional semantic space (T-Space) for linked datasets [121]. The T-Space is built using a distributional semantic model based on statistical semantic information derived from Wikipedia. This model enables flexible semantic matching in the search

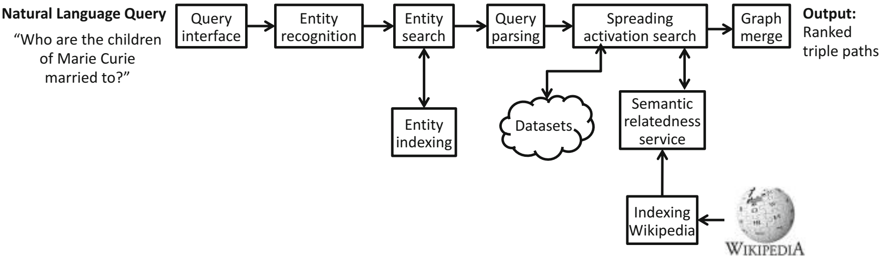


**Fig. 7.6** High-level architecture components for Treo (best-effort natural language) [111]

process. The definition of the T-Space provides a principled representation of datasets focused on addressing the expressivity–usability trade-off.

### 7.4.3 Discussion

Table 7.1 lists how each category addresses the key requirements for search and query over heterogeneous knowledge graph within a linked dataspace. The practical relevance of a search and query mechanism lies in the fact that structured data is a fundamental component of data sources where the effort associated with accessing this structured data is still significant and heavily mediated by database experts. The dissolution of the expressiveness/usability trade-off is the goal of schema-agnostic query approaches (see Fig. 7.7) that provide a semantic matching approach which enables the alignment or semantic mapping of the data consumer's query to the database conceptual model elements. Based on the analysis in Table 7.1, we can see that the Treo approach meets most of the requirements identified. Best-effort natural language search approaches provide a robust semantic matching approach. However, they relax expectations in terms of query results, delegating the results' final assessment to end users.

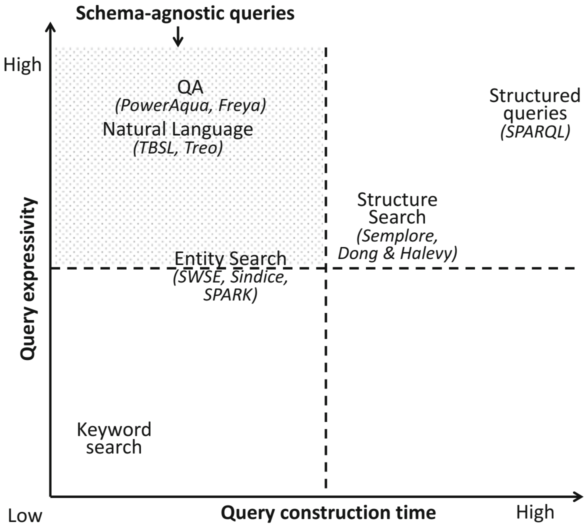


**Fig. 7.7** Query expressivity vs. Query construction time quadrant. Schema-agnostic queries allow both high expressivity and low query construction time

**Table 7.1** Coverage of approaches to address dataspace query requirements

| | Query requirements | | | | | | |
|---|---|---|---|---|---|---|---|
| Approaches | Level of schema agnosticism | High query expressivity | High scalability | Interactive search and low query execution time | Accurate and comprehensive semantic matching | Low setup and maintainability effort | High usability and low query construction time |
| *Information retrieval* | | | | | | | |
| Entity-centric: SWSE/ Visinav [164] | − | +− | ++ | ++ | − | ++ | +− |
| Entity-centric: Sindice/ Sigma [165] | − | +− | ++ | ++ | − | ++ | +− |
| Entity-centric: SPARK [166] | + | +− | NE | NE | + | ++ | ++ |
| Structure indexes: Semplore [167] | +− | +− | ++ | ++ | − | ++ | +− |
| Structure indexes: [168] | +− | +− | + | ++ | + | ++ | ++ |
| *Natural language approaches* | | | | | | | |
| QA systems: PowerAqua [169] | + | + | + | + | + | + | ++ |
| QA systems: Freya [170] | + | + | NE | + | + | − | + |
| Natural language search: TBSL [172] | + | + | + | + | + | + | ++ |
| Natural language search Treo [173], Treo T-Space [121] | ++ | ++ | + | + | ++ | ++ | ++ |
| *Structured queries* | | | | | | | |
| SPARQL | − | ++ | ++ | NA | NA | −− | −− |

++ requirement dimension is well covered
+ requirement dimension is partially covered with positive
+− there is an attempt to address requirement dimension, but the solution is not effective
− requirement dimension is poorly covered
−− requirement dimension is very poorly covered
NA requirement dimension is not addressed or focused on the research
NE dimension dependent on evaluation is either poorly or not evaluated

## 7.5 Design Features for Schema-Agnostic Queries

Leveraging existing work [111] we analysed the current approaches to determine the design features present in search and query mechanisms over heterogeneous data to determine the key features needed for a knowledge graph query mechanism for RLDs. The result of this analysis is presented in Table 7.2, where we can see five key design features emerging as clear trends for the creation of search and query services over heterogeneous knowledge graphs [111].

**Query Type** Entity-centric search, keyword-based search, natural language queries, and structured SPARQL queries represent complementary search and query services that might suit users in different tasks and purposes. Search and query platforms should explore this complementary aspect regarding heterogeneous data to enable users to switch among different search and query strategies. SWSE and Sindice explore this trend; however, the availability of natural language queries is a key feature not present in these systems. As part of the search and query features, users should be able to explore, understand, and refine search results by relying on navigational, browsing, and filtering capabilities integrated into the process (this functionality is present in SWSE, Sindice, and Semplore).

For many years, the difficulties associated with the hard constraints of the question answering problem have overshadowed the potential for applying NLP techniques for queries. NLP has developed a large set of techniques and tools for parsing and analysing users' information needs expressed as natural language queries. Different flavours of syntactic parsers, morphological analysers, and named entity recognition techniques are widely and effectively employed in QA systems and natural language search interfaces (e.g. PowerAqua, FREyA, Treo, and Treo T-Space). Recently, the efficacy of NLP techniques was demonstrated in the IBM Watson system [174], which outperformed a human contestant in a "Jeopardy" challenge. Watson heavily leverages standard NLP techniques to build a complex information extraction and search pipeline. Search and query mechanisms can explore NLP techniques to provide expressive and intuitive query interfaces.

**Disambiguation** The presence of ambiguity and incomplete information is intrinsic to the search and query process. As already explored in systems such as FREyA and Semplore, user feedback can help resolve ambiguities, enrich an application's semantic model, and filter and post-process results. Providing a supporting context around the answers can help users assess the data's correctness. In the Treo approach, the path in the dataset generated during the querying process provides contextual information for users. A best-effort approach can live together with database operations, such as aggregations, via data filtering mechanisms that let users remove incorrect entries from the results (e.g. using the associated type information).

**Ranking** In "If You Have Too Much Data, then 'Good Enough' Is Good Enough" [76], Pat Helland summarises the mindset shift that must occur in heterogeneous and distributed data environments, where many still expect the accurate and crisp results

**Table 7.2** Design features of schema-agnostic query mechanisms for heterogeneous knowledge graphs

| | Design features | | | | | |
|---|---|---|---|---|---|---|
| Approaches | Query type | Disambiguation | Ranking | Semantic approximation | Supporting knowledge bases/ linguistic resource | Performance/ scalability mechanism |
| *Information retrieval* | | | | | | |
| Entity-centric: SWSE/Visinav [164] | Keyword/SPARQL | None | Modified TF/IDF + link-based | None | None | Inverted index |
| Entity-centric: Sindice/Sigma [165]) | Keyword/Star-shaped | None | Modified TF/IDF + link-based | None | None | Inverted index |
| Entity-centric: SPARK [166] | Keyword-based | None | Yes | Dataset term expansion/ edit distance, substring matching | WordNet | |
| Structure indexes: Semplore [167] | Keyword with structural information (single-atom queries, path queries, star-shaped queries, entity queries, and tree-shaped queries) | Facet-based | Yes | Dataset term expansion (taxonomical enrichment) | None | Inverted index (position index) |
| Structure indexes: [168] | Keyword with structure information | None | Yes | Query/dataset term expansion (synonyms) | WordNet | Inverted index |
| *Natural language approaches* | | | | | | |
| QA systems: PowerAqua [169] | Full natural language | None | Based on the similarity scores | WordNet-based (hypernym, hyponym, synonym) semantic/ | WordNet | NA |

(continued)

**Table 7.2** (continued)

| | Design features | | | | | |
|---|---|---|---|---|---|---|
| Approaches | Query type | Disambiguation | Ranking | Semantic approximation | Supporting knowledge bases/ linguistic resource | Performance/ scalability mechanism |
| | | | | string similarity, based on taxonomical relations in the data | | |
| QA systems: Freya [170] | Full natural language | Disambiguation dialog | Yes | Query/dataset term expansion, manual lexicon enrichment | WordNet | NA |
| Natural language search: TBSL [172] | Full natural language | | Yes | Corpus pattern mining | WordNet, Corpus-based | Yes |
| Natural language search: Treo [173], Treo T-Space [121] | Full natural language | User feedback | Based on distributional semantic relatedness measure | Context-based distributional semantic approximation | Wikipedia-based ESA model | Distributional-relational structured vector space model |

typical for siloed databases. This trade-off is discussed in more detail in Chap. 3. The challenge of building query solutions with high usability and expressivity in a dataspace is coping with the data's semantic heterogeneity; this demands to relax our expectations of the results into a best-effort solution. Ranked lists of results in which users can assess those results' suitability are widely used in document search engines; web users have been extensively exposed to this approach and are thus familiar with best-effort search models. However, although document search engines can potentially return a long list of candidate documents, the best-effort query [171, 175] and ranking [176, 177] mechanisms for dataspaces should leverage the structure and types present in the data to target more concise answer sets. A number of dataspace search and query approaches leverage associations and relations to rank results [87, 168, 178–180].

**Semantic Approximation** The difficulty in effectively providing a robust semantic matching solution has been associated with a level of semantic interpretation that depends on fundamental and hard problems in artificial intelligence, such as common-sense knowledge representation and reasoning. Dataspace query approaches have considered both synonyms [181] and similarity [182] within the matching process. Recently, distributional semantic approaches have emerged as solutions to provide robust semantic matching by leveraging the use of semantic information embedded in large amounts of web corpora.

Distributional semantic models assume that the context surrounding a given word in a text provides essential information about its meaning [183]. Distributional semantics focus on constructing a semantic representation of a word based on the statistical distribution of word co-occurrence in texts. The availability of high-volume and comprehensive web corpora has made distributional semantic models a promising approach for building and representing meaning. However, the simplification of distributional semantic models implies some constraints on its use as a semantic representation. Distributional semantic models are suitable for computing semantic relatedness, which can act as a best-effort solution for providing robust semantic matching solutions for linked data queries (present in the Treo T-Space system).

**Supporting Knowledge Bases/Linguistic Resources** The availability of large amounts of unstructured text and structured data on the web can help to bootstrap a level of semantic interpretation based on available open and domain-specific knowledge. It is possible to address the volume of unstructured text corpora necessary to build distributional semantic models by using comprehensive knowledge sources available on the web, such as Wikipedia (present in the Treo and Treo T-Space systems). In addition, it is possible to use the semantically rich entity structure of data sources such as DBpedia (http://dbpedia.org), YAGO (www.mpi-inf.mpg.de/yago-naga/yago/), and Freebase (www.freebase.com) as a general-purpose entity and entity typing system that can easily integrate to the target datasets to provide a minimum level of structured common-sense knowledge, and which can later be used to improve semantic interpretation and tractability. RDF's standardised graph-based format facilitates the reuse and integration of existing data sources into target datasets.

## 7.6 Summary

The emergence of heterogeneous and distributed data environments such as the web of data, knowledge graphs, and dataspaces, in contrast to small controlled schema databases, fundamentally shifts how users search and query data. Approaches used for searching and querying siloed databases fail within these large-scale heterogeneous data environments because users do not have an a priori understanding of all the available datasets. This chapter investigates the main challenges in constructing a query and search service for knowledge graphs within a dataspace. The search and query services within a dataspace do not follow a one-size-fits-all approach and utilise a range of different techniques from keyword search to structured queries and question answering to support different characteristics of data sources, and user needs in the dataspace. Our analysis of the state of the art shows that existing approaches based on IR and natural language query interfaces have complementary design features, which, if combined, can provide schema-agnostics solutions to the usability and semantic matching challenges of querying large-scale heterogeneous data.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

# Chapter 8
# Enhancing the Discovery of Internet of Things-Based Data Services in Real-time Linked Dataspaces

Wassim Derguech, Edward Curry, and Sami Bhiri

**Keywords** Sensor indexing · Service discovery · Formal concept analysis · Dataspaces · Intelligent systems · Internet of Things

## 8.1 Introduction

A dataspace is an emerging data management approach used to tackle heterogeneous data integration in an incremental manner. Data sources that are participants in a dataspace can be of various types such as online services, open datasets, sensors, and smart devices. Given the dynamicity of dataspaces and the diversity of their data sources and user requirements, finding appropriate sources of data can be challenging for users. Thus, it is important to describe and organise data sources in the dataspace efficiently. In this chapter, we present an approach for organising and indexing data services based on their semantic descriptions and using a feature-oriented model. We apply Formal Concept Analysis for organising and indexing the descriptions of sensor-based data services. We have experimented and validated the approach in a real-world smart environment which has been retrofitted with Internet of Things-based sensors observing energy, temperature, motion, and light.

The chapter is structured as follows. Section 8.2 explores the need for discovery of data services within dataspaces. Section 8.3 provides an overview of existing semantic approaches to service discovery. Section 8.4 sets out the theoretical foundations of Formal Concept Analysis and shows how it can be applied to indexing Internet of Things-based data services based on their capabilities using a concept lattice. It then shows how the concept lattice can be used to discover sensors and relationships between sensor properties. Section 8.5 reports on a smart environment intelligent system validation of the discovery approach. Finally, Sect. 8.6 draws conclusions and details of future work.

© The Author(s) 2020

E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_8

## 8.2 Discovery of Data Services in Real-time Linked Dataspaces

With the trend of Industry 4.0 [184], the advent of the Internet of Things (IoT) [185], and the decreasing costs and increasing capabilities of sensors and smart devices, modern businesses are integrating more and more real-time data into their business processes [186]. New challenges facing modern business processes include the dynamic and efficient discovery of resources such as data sources and services [185]. Indeed, many business processes rely on IoT sensor data to provide the necessary business intelligence and insight to support decision-making. With the rapid adoption of IoT devices and sensors, the number of available sources for real-time data has risen. A key challenge for intelligent systems within IoT-based smart environments is to discover and select appropriate sources of real-time data.

### *8.2.1 Real-time Linked Dataspaces*

A data platform can provide a clear framework to support the sharing of data among a group of intelligent systems within a smart environment [1] (see Chap. 2). In this book, we advocate the use of the dataspace paradigm within the design of data platforms to enable data ecosystems for intelligent systems. A dataspace is an emerging approach to data management which recognises that in large-scale integration scenarios, involving thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources [2]. Within dataspaces, datasets *co-exist* but are not necessarily fully integrated or homogeneous in their schematics and semantics. Instead, data is integrated on an *as-needed* basis with the labour-intensive aspects of data integration postponed until they are required. Dataspaces reduce the initial effort required to set up data integration by relying on automatic matching and mapping generation techniques. This results in a loosely integrated set of data sources. When tighter semantic integration is required, it can be achieved in an incremental *pay-as-you-go* fashion by detailed mappings among the required data sources.

We have created the Real-time Linked Dataspace (RLD) (see Chap. 4) as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data, knowledge graphs, and real-time stream and event processing capabilities to support a large-scale distributed heterogeneous collection of streams, events, and data sources [4].

### *8.2.2 Data Service Discovery*

Within an RLD, creating explicit links between sensors-based data service descriptions helps to discover similar data services and consequently facilitate balancing observations from one sensor to the other. A possible use case can be a smart building within an energy management application where a decision support model is used to control the supply and demand of energy. A principal source of information used for decision support models within smart buildings is sensors. An efficient decision support model in such a context requires that the sensor data provided is correct and timely. However, faults may occur at any time. For example, a sensor may become unresponsive or start to produce low-quality data. In these cases, the discovery support service should provide suggestions for alternative sources of data. This can be quickly implemented if sensors are adequately described and organised in a semantically enriched and linked manner.

In our past work, we have investigated the use of Formal Concept Analysis (FCA) [187] to better organise a repository of services' descriptions in order to make their discovery more efficient [188]. In this chapter, we revisit this work to contextualise it within the dataspace paradigm.

## 8.3 Semantic Approaches for Service Discovery

Several existing works use semantic technologies and related solutions for the indexing and discovery of services based on their semantic descriptions [189–192]. A key challenge with using these tools for service discovery in highly dynamic and interactive environments, such as IoT-based smart environments, is their cost in terms of computational resources. The high computational costs of some solutions [189, 190] are their dependency on reasoning within the discovery algorithms, a time-consuming task. Other solutions [192] try to resolve this issue by providing off-line indexing mechanisms for reducing the time complexity of discovery algorithms. However, given the dynamicity of smart environments, the diversity of their features and user requirements, reconstructing the entire indexing structure is challenging within medium and large-scale environments. Therefore, indexing/organising capabilities to enhance their discovery is required.

This section examines approaches that propose to use indexing structures for enhancing the discovery of IoT data services (e.g. a sensor) in a given repository (e.g. a dataspace). For each of the approaches analysed, we consider the following two key requirements:

- *Requirement 1—Ontology-based Discovery*: Searching resources should rely on concepts from domain ontologies used in their descriptions without relying on keyword extraction from textual descriptions [193]. Relying on extracting keywords from unstructured textual descriptions can lead to inconsistent results [194]. This requirement was elicited from the following works: [193, 195], and from Semantic Web Services Models: WSMO [196] and OWL-S [197].
- *Requirement 2—Time Performance*: Searching for a service or a business process can often rely upon reasoning, which makes the discovery very slow [191]. The second requirement is a low-latency response within large repositories of services

or business processes. This requirement was elicited from the following works: [190–192].

Using these requirements, we analyse related work in terms of their indexing *mechanism*, underlying *service description language*, and *limitations*.

### 8.3.1 Inheritance Between OWL-S Services

The discovery of semantic web services is challenging within large repositories due to costly reasoning operations. One solution to this problem is to introduce inheritance between OWL-S services [198]. Their specification denotes the possibility to define service profiles' hierarchies similar to object-oriented inheritances. Inheritance relationships between services are proposed to find service substitutes by exploring services that are higher in the hierarchy. Similar to object-oriented concepts, a sub-service may be used to substitute its super-service for automated, dynamic service discovery and composition [198]. Inheritance is also useful for creating new service profiles as a subclass of an existing profile. This makes the new service inherit the properties defined in the superclass profile. Other approaches [189] propose to capture such hierarchies in a visual editor for an OWL-S service description editor without discussing how these hierarchies can be created.

Introducing inheritance was a natural choice to enhance service discovery operation [198]. Services and relations are defined in the OWL language, and consequently, fulfils *Requirement 1—Ontology-based discovery*. However, little research has been carried out to determine inheritance between web services [199]; consequently, we cannot further comment on *Requirement 2—Time performance*.

### 8.3.2 Topic Extraction and Formal Concept Analysis

The work carried out by Aznag et al. [190] investigated the use of formal concept analysis as an indexing tool using topics extracted from service descriptions using SA-WSDL. Starting from a set of service descriptions, their algorithm converts them into a "service transaction matrix" that captures for each service the relevant textual concepts used in its description. This matrix is further refined with probabilistic clustering of the textual concept in order to extract a set of topics. The result of this analysis generates the correlated topic model that holds for each service and the topics it belongs to, with specific probabilities. In the work, formal concept analysis is used exclusively for clustering the extracted topics to make discovery easier when using a concept lattice. The use of formal concept analysis in this approach is due to its broad adoption as a well-established mathematical theory of concepts and concept hierarchies, which makes the service discovery much easier [190].

Topics used in the concept lattice are textual concepts that are extracted from the textual description of services, and consequently, this approach does not fulfil *Requirement 1—Ontology-based discovery*. Concerning *Requirement 2—Time performance*, the authors did not perform any evaluation of the time required to create the cluster of services. Nevertheless, they indicate that the query response time varies between 300 and 3000 ms with a test collection of 1088 services. Given that the discovery operation using formal concept analysis is a simple tree parsing operation, it has a linear complexity depending on the number of concepts in the created lattice (tree). Furthermore, in formal concept analysis, the creation of the concept lattice is the most expensive operation [200]. Thus, this can lead to the conclusion that the construction time of the entire cluster could be in the order of seconds.

### *8.3.3 Reasoning-Based Matching*

Srinivasan et al. [192] looked into enhancing the indexing of a UDDI registry of services described in OWL-S during the service advertisement phase. Assuming that services are described using predefined ontologies, they use a matching degree between inputs and outputs of services. The matching uses concepts from the service description ontologies in order to identify a correct clustering of service descriptions into predefined ontological clusters similar to the North American Industry Classification System (NAICS) [201].

*Requirement 1—Ontology-based discovery* is fulfilled, as this approach relies exclusively on clusters that are constructed from hierarchical ontological concepts similar to NAICS [201]. However, the problem with this approach is that it relies heavily on reasoning and pre-computing information required for the search request, which is costly (*Requirement 2—Time performance*). The authors have confirmed this limitation through the performance evaluations that they carried out. The OWLS/UDDI approach takes more than 4000 ms for inserting 50 advertisements into the registry, which was 6–7 times slower than using a classical UDDI approach. However, the authors argue that this time is not very important as the advertisement operation can be done off-line and more time can be saved during the discovery phase without giving any quantifications.

### *8.3.4 Numerical Encoding of Ontological Concepts*

Mokhtar et al. [191] optimise the indexing of service descriptions by avoiding semantic reasoning and by using a numeric coding scheme, a widely adopted method for enhancing the performance of ontology processing. They propose that a service registry can be clustered using a predefined ontology or taxonomy such as NAICS

[201] or UNSPSC [202], where an interval of numbers encodes each ontological concept. These intervals are defined using a linear inverse exponential function in a way where one interval can be contained in another one without overlap, creating a subscription relation. For example, to model sub-concept relations between Wi-Fi and Wireless, Wi-Fi can be coded by the interval [0, 0.1] and Wireless by the interval [0, 1] (i.e. [0, 0.1] is contained in [0, 1]). One can add to this example another concept, Bluetooth, that is, a sub-concept of Wireless by assigning the interval [0.2, 0.3] to Bluetooth (such that [0.2, 0.3] is contained in [0, 1], where [0, 0.1] and [0.2, 0.3] do not overlap).

Similar to [192], the authors of [191] rely exclusively on clusters that are constructed from hierarchical ontological concepts similar to NAICS [201]. Consequently, *Requirement 1—Ontology-based discovery* is fulfilled.

Concerning *Requirement 2—Time performance*, and compared to the performance of the work by Srinivasan et al. [192], Mokhtar et al. [191] achieve better results as the required time for encoding and advertisement does not exceed 450 ms for 50 service descriptions. This performance is achieved with the assumption that the used ontologies for service classification are encoded. Similarly, the reasoning operation is reduced to a comparison of codes/intervals. In this case, to infer that a concept c1 subsumes another concept c2, one needs to evaluate if their corresponding encoding interval of c1 is contained in the encoding interval of c2. This restricts the system to use classification ontologies that do not frequently evolve. Otherwise, service advertisements and requests need to check and update their encoding intervals periodically.

Similarly, Binder et al. [203] looked into encoding techniques to create hierarchies of service descriptions. Here the authors used the Generalised Search Tree algorithm [204] for organising service descriptions. This technique provides efficient search in the order of milliseconds over 10,000 entities. The main limitation of this approach is the complexity of maintaining this indexing structure: a new entry requires around 3 s for updating the tree [191].

### 8.3.5 Discussion

In summary, most of the analysed approaches rely on indexing service descriptions using existing taxonomies such as the North American Industry Classification System (NAICS) [201], UNSPSC [202], or the MIT Process Handbook [205]. This supports the idea of using ontologies as a common conceptualisation and shared understanding among service providers, registry hosts, and service requesters. However, the use of these ontologies makes the indexing heavily reliant on reasoning, a task that can be costly. The literature proposes multiple techniques that either used reasoning or proposed alternative solutions.

**Table 8.1** Comparative analysis of capability indexing approaches

| Approach | R1: Ontology-based discovery | R2: Time performance | Limitations |
|---|---|---|---|
| Inheritance between OWL-S services [189] | Not fulfilled | N/A | No clear methodology on how a hierarchy is created. |
| Topic extraction and formal concept analysis [190] | Fulfilled | Size: 1088 services, query response time between 300 ms and 3000 ms | Topic extraction and correlations are based on a probabilistic system.<br>Solution needs further optimisations.<br>FCA is exclusively used for topic clustering. |
| Reasoning-based matchmaking [192] | Fulfilled | Size: 50 services, index construction + advertisement time: ~4 s | Service advertisement operation is costly and heavily relying on reasoning. |
| Numerical encoding of ontological concepts and codes comparison [191] | Fulfilled | Size: 100 services, index construction + advertisement time: ~500 ms | Requires periodic updates of the codes of the clustering ontology.<br>Slow registry maintenance for large repositories. |

The analysis is summarised in Table 8.1. The key findings are:

- Indexing or clustering of service descriptions using ontologies is widely adopted [190–192].
- Maintainability of the indexing structure is critical to the applicability of the proposed approach [191, 192].
- The benefits of reusing existing techniques such as FCA for creating or maintaining the indexing structure is widely accepted [190].

## 8.4 Formal Concept Analysis for Organizing IoT Data Service Descriptions

In our past work, we have investigated the use of Formal Concept Analysis [187] to better organise a repository of service descriptions in order to make their discovery more efficient [188]. In this section, we revisit this work to contextualise it within the dataspace paradigm. We start by defining the theoretical foundations of FCA while applying it to IoT sensor data service descriptions.

FCA is a technique that has evolved from mathematical lattice theory and has been used for data analysis across several domains, such as organising web search

**Table 8.2** Data table with binary attributes for sensor-based data services [188]

| Objects | Active | Storage option | Digital display | Accessible |
|---|---|---|---|---|
| Sensor 1 | X | X | X | X |
| Sensor 2 | X |  | X | X |
| Sensor 3 |  | X | X | X |
| Sensor 4 |  | X | X | X |
| Sensor 5 | X |  |  |  |

results into concepts based on common topics, gene expression data analysis, information retrieval, and understanding and analysis of source codes [206]. FCA helps in identifying meaningful relationships within a set of objects that share common attributes. It also provides a theoretical model to build from a *formal context* a partially ordered structure called a *concept lattice*.

### 8.4.1 Definition: Formal Context

A *formal context FC* is a triplet <*X,Y,R*>, where *X* and *Y* are non-empty sets and $R \subseteq X * Y$ is a binary relation between *X* and *Y*.

For a formal context *FC*, elements $x \in X$ are referred to as objects and elements $y \in Y$ are called attributes. $<x,y> \in R$ denotes that the object *x* has the attribute *y*.

In this work, the formal context is defined via the set of sensor data services as well as their respective descriptions. Table 8.2 will be used in this section as a running example that describes the relationships between the objects (i.e. sensors 1–5 represented by the table rows: *X* =*{Sensor 1, Sensor 2, Sensor 3, Sensor 4, Sensor 5}*) and their descriptions (i.e. attributes represented by the table columns: *Y* =*{Active, Storage Option, Digital Display, Accessible}*, in Table 8.2). This example considers the following four attributes:

- *Active:* Indicates if the sensor is in operation
- *Storage Option:* Indicates if the sensor can store data
- *Digital Display:* Indicates if the sensor is equipped with a digital display for displaying the data.
- *Accessible*: Indicates if the sensor is located in an accessible area

Another fundamental concept in FCA is the *Formal Concept*.

### 8.4.2 Definition: Formal Concept

A *formal concept* in <*X,Y,R*> is a pair <*E,I*> of $E \subseteq X$ (called *extent*) and $I \subseteq Y$ (called intent) such that *Att(E)* = *I* and *Obj(I)* = *E*. *Att(E)* is an operator that assigns subsets of *X* to subsets of *Y*, such that *Att(E)* is the set of all attributes shared by all objects from *E*. *Obj(I)* is an operator that assigns subsets of *Y* to subsets of *X*, such that *Obj(I)* is the set of all objects sharing all the attributes from *I*.

From this definition, one can conclude that a concept *C* = *<E,I>* is created by getting objects from *E* sharing the same attributes from *I*. For example, the shaded rectangle in Table 8.2 represents a formal concept *<E_1,I_1>* = *<{Sensor 1, Sensor 2, Sensor 3, Sensor 4}, {Digital Display, Accessible}>* because *Att(E_1)* = *{Digital Display, Accessible}* and *Obj(I_1)={Sensor 1, Sensor 2, Sensor 3, Sensor 4}*.

From a *formal context FC* = *<X,Y,I>*, one can deduce a set of formal concepts that can be ordered with respect to a *sub-concept ordering*.

### *8.4.3 Definition: Sub-concept Ordering*

Having two formal concepts *<E_1,I_1>* and *<E_2,I_2>* from *FC* = *<X,Y,R>*, *<E_1,I_1>* ≤ *<E_2,I_2> iff E_1* ⊆ *E_2* (*iff I_2* ⊆ *I_1*).

Let us consider the following formal concepts from the example in Table 8.2:

*<E_1,I_1>* = *<{Sensor 1, Sensor 2, Sensor 3, Sensor 4}, {Digital Display, Accessible}>*
*<E_2,I_2>* = *<{Sensor 1, Sensor 2, Sensor 4}, {Digital Display, Accessible}>*
*<E_3,I_3>* = *<{Sensor 1, Sensor 2}, {Active, Digital Display, Accessible}>*
*<E_4,I_4>* = *<{Sensor 1, Sensor 2, Sensor 5}, {Active}>*

Then

*<E_3,I_3>* ≤ *<E_1,I_1>*, *<E_3,I_3>* ≤ *<E_2,I_2>*, *<E_3,I_3>* ≤ *<E_4,I_4>* and *<E_2,I_2>* ≤ *<E_1,I_1>*.

The set of ordered formal concepts derived from a formal context is called a *concept lattice*, which is another important notion in FCA. A concept lattice can be represented as a graph such as the one depicted in Fig. 8.1 (created using Conexp

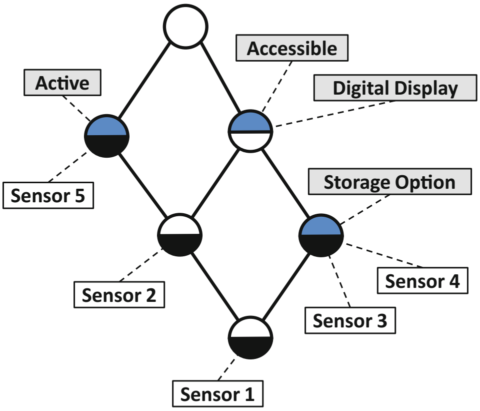


**Fig. 8.1** Concept lattice of the example in Table 8.2 [188]

[207]). In this figure, the concept extent near the bottom of the lattice contains only *Sensor 1* since the corresponding intent is related to the most significant number of attributes. The top concept contains all the sensors, and its intent corresponds to no attribute. This makes the concept less attractive as it allows for all possible combinations of attributes.

In this example, we considered only binary attributes (i.e. either the object has or does not have that attribute). However, in real settings, when describing services, there are also multi-valued attributes that need to be transformed into a binary attribute through a scaling operation. For details about the process, we refer the reader to our publication, which details the scaling operation applied for this example [188].

This concept lattice is an indexing structure; it allows the organisation of sensor-based data services descriptions in a tree. This structure captures the explicit relations between formal concepts. This is useful to discover data services that share similar attributes. For example, if one of the sensors is not active anymore, it is possible to use one of the other sensors in its equivalence class or discover sensors that share its attributes. Furthermore, the presence of the explicit sub-concept relationship between concepts allows the discovery of additional knowledge among the objects' attributes that are analysed (i.e. sensor attributes). Indeed, as depicted in Fig. 8.1, one can discover implications such as: every sensor that has a "Storage Option" is also "Accessible" and has a "Digital Display". In other words: "Storage Option" implies "Accessible" and "Digital Display". Algorithms for knowledge discovery and implications have been presented in previous work [188].

It is important to note that the use of FCA permits the creation of a concept lattice uniformly. In other words, it always creates the same structure with the same input objects. This has the advantage of creating a deterministic discovery algorithm, as there is no need to use any heuristic for parsing this indexing structure. This chapter focuses mainly on the creation of the concept lattice and the study of its applicability for indexing a set of IoT sensor data service capabilities in a dataspace. In the next section, we use FCA in a real-world intelligent system setting for organising data service descriptions for a smart environment.

## 8.5 IoT-Enabled Smart Environment Use Case

This section illustrates a smart environment use case using a set of real-world IoT sensors deployed using the RLD where energy-related data is made available and interlinked to support intelligent system decision-making and ultimately improve energy consumption behaviour [100]. The RLD has been realised within a number of smart environments from smart homes to smart airports as detailed in Chap. 14. The data from the smart environments is provided by real-time data sources such as IoT sensors as well as relatively static background knowledge such as the building plan and occupancy.

In this section, we examine an intelligent system for energy management within a smart building pilot where there are various sources of power consumption, including heating, ventilation, and air conditioning (HVAC) systems, lights, and electronic devices. The building has been retrofitted with energy sensors to monitor the consumption of power. In total, there are over 50 fixed energy consumption sensors covering office space, a cafe, a data centre, kitchens, conference and meeting rooms, a computing museum along with over 20 mobile sensors for devices, light, temperature, and motion detection.

A building-specific deployment of the RLD has been presented in [62] with a sensor network-based situation awareness scenario presented in [208]. In total, this work used a total number of 78 sensors. The resulting IoT sensor data services are described via the following set of attributes:

- *Active:* This attribute reports whether the sensor is in operation.
- *Observed Phenomenon:* We have four observed phenomena, which are “energy and power consumption”, “motion”, “light”, and “temperature”. This attribute is a multi-valued attribute that needs to be scaled.
- *Protocol:* This attribute indicates the protocol used by the sensor. We have in our selection of sensors two possible protocols: UDP used by electricity and power consumption sensors and CoAP used by other sensors. This is a multi-valued attribute that has to be scaled.
- *Electricity Phases:* This attribute reports on the electricity phases used by the sensor; we have two options: three-phases and one-phase sensors. Again, this is a multi-valued attribute that has to be scaled.
- *Location:* Even though this attribute is not an intrinsic property of the sensor, we have used it because it is important information that is required for processing the data provided by the sensor. This is also a multi-valued attribute that enumerates the locations of the sensors, for example, 1st floor: west wing, ground floor: canteen, which needs to be scaled.

All the sensor data service capabilities were automatically generated from a tabular file containing the original descriptions that were manually checked. Manually checking RDF descriptions was possible as the number of sensors used was limited. We have not carried out any evaluation of the developed RDF parser, because it is custom made for the dataset and conceptual model. The correctness of the algorithm we applied for the RDF parser is out of the scope of this chapter; however, the data has been manually verified after parsing and scaling.

The resulting concept lattice from Conexp [207] is depicted in Fig. 8.2. The top concept in this lattice represents the set of all active sensors <*{Sensor 1, Sensor 2, ... Sensor 78}, {Active}*>. This formal concept contains in its extent all the sensors of the dataset because they are all active. One can see in this concept lattice several formal concepts that represent the set of motion sensors <*{Sensor 61,... Sensor 66}, {OP: Motion}*>, the formal concept for temperature sensors <*{Sensor 67,... Sensor 72}, {OP: Temperature}*> and the light sensors <*{Sensor 73,... Sensor 78}, {OP: Light}*>. These three formal concepts are sub-concepts of the concept <*{Sensor*

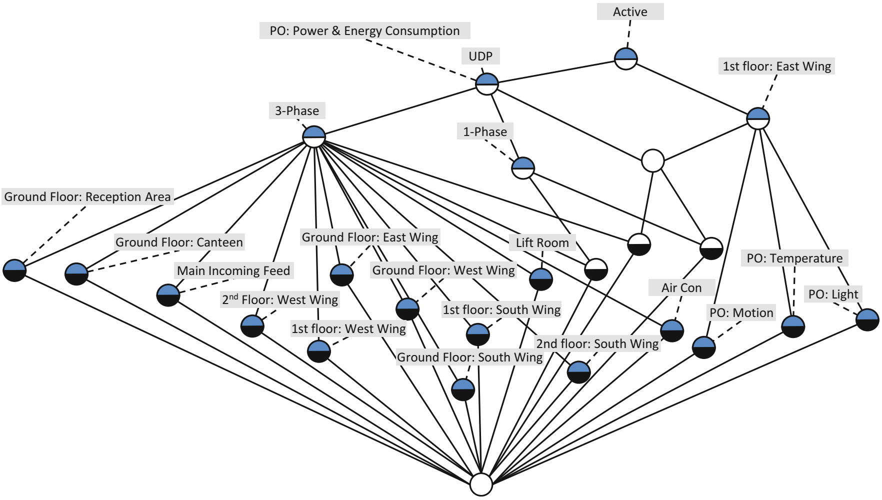


**Fig. 8.2** Concept lattice of the smart office use case [188]

**Table 8.3** Comparing time performances of indexing approaches

| Indexing mechanism | Time performance |
|---|---|
| Inheritance between OWL-S services [189] | N/A |
| Topic extraction and FCA [190] | Size: 1088 services, query response time between 300 and 3000 ms |
| Reasoning-based matchmaking [192] | Size: 50 services, index construction + advertisement time: ~4 s |
| Numerical encoding of ontological concepts and codes comparison [191] | Size: 100 services, index construction + advertisement time: ~500 ms |
| Capabilities indexing using FCA [188] | Size: 1000 capabilities, index construction + parsing time: ≤25 ms |

*61, ... Sensor 78}, {1st Floor:East Wing}>*. This helps to infer that all motion, temperature, and light sensors are in the same location (1st Floor: East Wing).

The focus of this work is on applying FCA in smart environments where the number of concepts does not exceed 1000. In such settings, the maximum construction time reaches only 25 ms [188]. Even though the main criticism towards using FCA, in this case, is the fact of reconstructing the concept lattice for any change in the environment (e.g. a new sensor, change in a sensor attribute), this remains acceptable with such low construction time.

Comparing the efficiency of the approach with respect to the approaches analysed in Sect. 8.3, we refer to Table 8.3. Each line of this table recalls the approach used, the indexing mechanisms, and the time performance as indicated by the authors in their papers. The table shows clearly that our approach outperforms the others because it does not use any reasoning for indexing the set of input capabilities.

## 8.6 Conclusions and Future Work

In this chapter, we discussed the use of Formal Concept Analysis (FCA) for indexing data sources, more specifically, IoT sensor data services in the context of dataspaces. Each of the data sources/services is described using a capability model capturing its functional and non-functional features. We use these descriptions for indexing the sensors using FCA, and the resulting indexing structure (called a concept lattice) which can support several use cases (e.g. the discovery of a replacement sensor).

In formal concept analysis, a formal context considers only single-valued attributes. In the case of multi-valued attributes, a scaling operation is required for transforming them into multiple single-valued attributes. In this chapter, we used only simple types of service capabilities that can be easily scaled from multi-valued to single-valued attributes. However, the model permits the modelling of complex values such as range and conditional values. Investigating scaling operations for covering these complex attribute types is required in order to consider them in the indexing approach using FCA.

Furthermore, a major concern in using FCA is its application for the analysis and indexing of large numbers of data services. FCA is known to be a memory and compute heavy technique [209]. In small-scale scenarios such as the use case in this chapter, the performance factor can be ignored as the computation time can be insignificant. However, in very large-scale deployments, this approach would ultimately fail because the time required to identify the concepts and create the lattice may be several hours. Incremental concept lattice creation can help in this direction. Indeed, FCA researchers [210] propose a concept lattice creation algorithm that has a quadratic worst-case time complexity in terms of the number of concepts, which could be leveraged in future work.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

# Chapter 9
# Human-in-the-Loop Tasks for Data Management, Citizen Sensing, and Actuation in Smart Environments

**Umair ul Hassan and Edward Curry**

**Keywords** Human-in-the-loop · Crowdsourcing · Citizen sensing · Task routing · Smart environments · Dataspaces · Smart environments

## 9.1 Introduction

Humans are playing critical roles in the management of data at large scales, through activities including schema building, matching data elements, resolving conflicts, and ranking results. The application of human-in-the-loop within intelligent systems in smart environments presents challenges in the areas of programming paradigms, execution methods, and task design. This chapter examines current human-in-the-loop approaches for data management tasks, including data integration, data collection (e.g. citizen sensing), and query refinement. A comparison of approaches (Augmented Algorithms, Declarative Programming, and Stand-alone Platforms) that can enable human tasks within data management is presented. The chapter also covers spatial tasks where users within the smart environment are requested to take physical actions in the environment in the form of citizen actuation.

This chapter discusses the design of the human task service and its use within intelligent applications in the Real-time Linked Dataspace (RLD). The rest of this chapter is organised as follows: the concept of the "Wisdom of the Crowds" and crowdsourcing is explained in Sect. 9.2. Section 9.3 examines the challenges to enable crowdsourcing, including issues with task design, task assignment, and user incentivisation. Section 9.4 introduces existing approaches to utilising humans-in-the-loop with comparisons provided in Sect. 9.5. Section 9.6 discusses the human task service of the RLD with emphasis on service-levels, applications, data processing pipeline, data model, and task routing. The chapter concludes with a summary in Sect. 9.7.

© The Author(s) 2020
E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_9

## 9.2 The Wisdom of the Crowds

Crowdsourcing has emerged as a powerful paradigm for solving complex problems at a large scale with the help of a group of people [211–213]. The rapid development of web technologies has made it possible for millions of online users to participate in collective efforts or "crowds" formed in response to open calls. People can contribute by performing tasks such as collecting photos, transcribing audio, classifying images, and classifying items [211]. The notion of "wisdom of crowds" advocates that potentially large groups of non-experts can solve complex problems usually considered to be solvable only by experts.

Crowdsourcing has been applied to develop prediction markets, design innovative solutions, and support knowledge generation, with the help of dedicated collaboration platforms such as Wikipedia or online workers to perform micro-tasks through marketplaces such as Amazon Mechanical Turk (AMT). Crowdsourcing would seem like a natural fit to be applied for supporting collaborative data management at large scales where human intelligence can complement machine computation to achieve higher-quality results and capitalise on the domain knowledge of the crowd. The suitability of crowdsourcing for solving complex problems has been demonstrated through several research and industrial projects [214, 215]. In a similar direction, the database research community has started to develop tools and techniques to ease the development efforts required to allow algorithmic access to crowds [216, 217].

While crowdsourcing focuses on combining the effort of a group of human contributors, *human computation* is a paradigm that focuses on an algorithmic approach towards harnessing human affordances [218, 219]. In practice, human computation can leverage crowdsourcing by asking a large number of people to perform specific computational tasks. Quinn and Bederson [218] define human computation based on two aspects of the problem at hand: first, it must follow the general paradigm of computation so that it can be solved by computers alone someday, and second, a computer controls the computational process. Finally, both of these mechanisms can be used to implement human-in-the-loop approaches where both human and machine intelligence are leveraged together. A common example of a human-in-the-loop system is to create machine learning models where humans are directly involved in training, tuning, and testing data for a machine learning algorithm. Crowd correct inaccuracies in machine predictions, thereby increasing accuracy, which results in a higher quality of results. Another common case is to include humans within the feedback-loop of automated decision-making or control systems system.

To demonstrate the application of human-in-the-loop and crowdsourcing, consider the example of an analyst who wants to prepare a list of craft shows, fairs, and festivals ranked according to variety. Instead of searching on the web and sifting through a plethora of websites, the analyst decides to crowdsource this data collection activity. After defining the required attributes of data, such as name, city, URL, and craft variety, the analyst posts a task on Amazon Mechanical Turk. Once the data collection tasks are complete, the analyst tries to de-duplicate entries with the help of a state-of-the-art algorithm. The algorithm is supplemented with the de-duplication performed by an expert to achieve high-quality results by enabling human tasks. Within the context of data management, human tasks have been demonstrated in different stages of data processing pipeline including the collection and enrichment of data, mapping and matching between schema and records, and feedback and refinement of the results of data quality algorithms.

Within the physical world, crowdsourcing techniques can be used to go beyond data management tasks to ask users to perform real-world "citizen actuation" tasks where users are requested to perform tasks to make a physical change in the environment [220]. The key to enabling the range of human-in-the-loop tasks is the use of a platform to manage your crowd of users.

### *9.2.1 Crowdsourcing Platform*

A crowdsourcing system, in general, has three types of interacting agents: requesters, workers, and platform. Each of these agents is described as follows:

- **Requesters:** Submit tasks to the platform that need to be performed by the crowd. Apart from humans, the requester can also be another application that needs human services for performing its functionality. Requesters are interested in maximising their utility, which is defined in terms of the quality of task performance and the associated costs. Note that the notion of quality and costs can vary between types of tasks and the application domain.
- **Workers:** Members of the crowd who are willing to perform tasks. Workers can vary in terms of their reliability of performed tasks and the incentive they expect for the work. The worker is interested in maximising their utility, which is defined in terms of the effort they exert and the value they gain from performing tasks.
- **Platform:** Serves as the mediator between requesters and workers; therefore, providing the interaction mechanism between both agents. It defines the mode of exchange for tasks, results, feedback, and incentives. A third-party platform provider is interested in maximising the value gained from the use of the software and its functionality. Furthermore, it is in the interest of platform managers to promote the long-term use of their platform.

Figure 9.1 highlights the sequence of interactions between these agents. (1) The requester submits tasks to the platform, which allows filtering of workers based on their characteristics or categories. (2) The tasks are assigned to the appropriate workers. (3) The workers perform the tasks and submit the responses to the platform. The platform assembles the results of crowdsourcing by aggregating and filtering the responses depending on the application domain. (4) The results are sent back to the requesters and (5) feedback on the performance of the workers is shared with the worker assignment component.

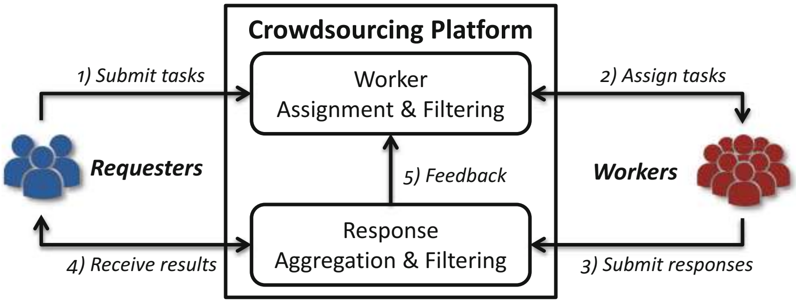


**Fig. 9.1** An overview of a typical interaction between agents in a crowdsourcing scenario [221]

The next section provides a short overview of technical challenges that are required to be addressed for enabling human tasks within smart environments.

## 9.3 Challenges of Enabling Crowdsourcing

The challenges of crowdsourcing entail considerations that are fundamental to enabling contributions from human participants in a smart environment. Law and Ahn [219] provide a detailed overview of human computation in general and its associated challenges, and Li et al. [222] provide a more specific review of the literature on crowdsourcing and data management.

**Task Specification** The first step towards enabling human-in-the-loop is to define the human tasks in terms of input, expected output, and constraints such as conditions for success and time limits. The flexibility of formalisms used for task specification varies among the existing research proposals from declarative methods to algorithmic code [223]. The specification of human tasks also depends on the systems responsible for their execution. For instance, an organisation can deploy a web-based API for allowing different services and applications to access human task services uniformly. Besides the basic specification of a task, additional details can include the details of the composition of a complex task into small tasks and their execution in a workflow [224].

**Interaction Mechanism** Interaction between a human and a human-in-the-loop process requires appropriate user interfaces on a variety of user devices to support the process. It is difficult to design one user interface that fits the requirements of all the various kinds of human tasks [225]. In this regard, existing approaches focus on generating task-specific user interfaces on-the-fly or using templates for different tasks [219]. Besides the design of the element to be presented on a device's screen, there are other factors of interaction mechanism that require careful consideration. For instance, how users will be notified when new tasks need their attention or what interaction mechanisms exist for users looking to perform available tasks. Human tasks can also be crowdsourced if the number of tasks is sufficiently large, and when the human roles have been defined to perform certain types of tasks. The choice of a crowdsourcing platform requires a trade-off between ease of implementation versus control over worker behaviour. Due to the limited choice of commercial crowdsourcing platforms, most of the existing research prototypes use Amazon Mechanical Turk for crowdsourcing micro-tasks [226]; however, there have been some research projects that also utilised open source platforms such as PyBossa [227].

**Task Assignment** Managing different types of tasks and assigning them to appropriate people is a fundamental challenge of crowdsourcing. The most straightforward approach, as followed by the majority of commercial crowdsourcing platforms, is to allow people to self-assign tasks by searching and browsing through appropriate user interfaces [211]. The alternative approach is to actively make assignment decisions algorithmically, an approach which is better suited for the objectives of human computation [211]. However, the random assignment of tasks to users may not be an effective strategy in domain-specific tasks. For example, domain experts are more suited to accomplish knowledge-intensive tasks that require specific expertise, which may not be available among users of general crowdsourcing platforms [228, 229]. In addition to expertise, the reliability [230] of people in performing tasks, and their physical location [212], can be taken into consideration when matching tasks with workers.

**Result Aggregation** The uncertainty of the correctness of results generated by human computers is an important challenge for enabling human tasks. A well-known solution to this challenge is to employ multiple human computers to perform the same task to ensure the quality of the results. Due to the limited availability of ground truth, the core challenge is to determine whether to accept or reject results generated by a human computer. If we accept results from multiple humans, then the challenge becomes one of how to combine them to generate an aggregated result. Existing approaches to data aggregation from crowdsourced human tasks range from expectation-maximisation algorithms [231] to probabilistic graphical models [232].

**Incentives Mechanisms** Motivating people to participate in human tasks is a challenge for system developers and researchers [233]. Although the accessibility problem has been alleviated due to the plethora of communication tools available for interacting with users over the Internet, the benefit of performing small units of work can be insignificant for the majority of users. The design of appropriate incentive mechanisms is an active area of research in human-in-the-loop and crowdsourcing. While monetary rewards are shown to be most effective in attracting the attention of people [233], other possible solutions include gamification [234] and altruistic motivations [235].

**Latency Issues** Machines and humans differ in the speed of work they can perform in a limited amount of time, which raises the issues of latency in human tasks. The availability of people who have the required knowledge or skill set for a task can also become a bottleneck, requiring changes to computational execution plans [236–

238]. Some efforts have shown that people can be paid to create pools of available crowd workers on-demand [239, 240]; such approaches can be used to support the crowdsourcing of low-latency data [241].

## 9.4 Approaches to Human-in-the-Loop

This section provides an overview of the human-in-the-loop approaches that have been organised into three high-level categories depending on their specialisation of database operations and generality of supporting different human tasks. Some representative examples of existing research are provided for each approach to illustrate the range of applications of human tasks in databases and integration systems.

### *9.4.1 Augmented Algorithms and Operators*

The primary idea of augmented algorithms and operators is to support the data transformation and analytical algorithms with human intelligence. Existing solutions in this category range from database operators (e.g. joins, sorts, skylines) to data integration processes (e.g. schema matching, entity resolution) to data quality updates (e.g. missing values, dependency resolution) [217]. For unstructured data, the algorithmic approaches support natural language processing in activities such as entity extraction and language translation [242]. For multimedia, human computation has been shown to complement a wide range of pattern recognition and machines learning algorithms [231].

Augmented algorithms have been used within dataspaces and databases for feedback-based refinement for data integration. For instance, Roomba is an iterative approach for matching data elements in a dataspace, which solicits human feedback on potential matches [118]. DSToolKit refines schema mappings by asking users to indicate the relevance of query results generated by possible mappings [243]. FICSR generates data constraints from possible schema alignments through an exploration process with the help of experts [244]. Van Keulen and de Keijzer proposed to iteratively reduce the number of possible alternatives in probabilistic data integration processes by soliciting user feedback on ranked query results [245].

The subjective nature of data quality also requires human validation in the data cleaning process. For instance, the GDR system takes a decision-theoretic approach for guiding database repairs through human feedback, and it employs active learning to capture human knowledge for further reduction of user workload [246]. The KATARA system leverages human computation to repair data in a table using a knowledge base [247]. The CrowdAidRepair system combines rules-based and crowd-based data repairing approaches interactively [248].

### 9.4.2 Declarative Programming

Declarative approaches focus on using human computation with the help of well-defined data operations. This approach facilitates independence with respect to the platform used to access human services with the flexibility to access such services within query or programming languages. Extending existing query languages, such as SQL, helps to minimise the learning curve associated with programming human computation. For instance, Deco [223], Qurk [249], and CrowdDB [250] extend SQL to provide database services on top of crowdsourcing platforms. Qurk uses user-defined functions to crowdsource relational comparisons and missing data. Likewise, CrowdDB introduces new SQL keywords to do the same. CrowdDB also allows data collection by defining annotations for columns and tables. Deco focuses on data collection with crowds through the definition of data fetch and resolution rules. Deco separates logical and physical tables to retain crowdsourced raw data. By contrast, hLog is a declarative language to specify human computation during the execution of information extraction and integration programs [251]. hLog extends Datalog with rules to specify details of the required human efforts in information extraction and integration programs. hLog also maintains provenance information to distinguish between crowd-generated versus computer-generated data. The Event Crowd [241] is a hybrid crowd-enabled event processing engine that uses five event crowd operators (Annotate, Rank, Verify, Rate, and Match) that are domain and language independent and can be used by any event processing framework. These operators encapsulate the complexities to deal with crowd workers and allow developers/data scientists to define an event–crowd hybrid workflow.

### 9.4.3 Generalised Stand-alone Platforms

This category of approaches focuses on building platforms with human-in-the-loop functionality, thus providing human intelligence services to other applications in a dataspace. These approaches do not depend on external platforms for human services as compared to previous approaches. For examples, Freebase was a web-based knowledge base that aimed to describe world entities with the help of crowdsourcing. It was supported by a human computation platform called RABj, which allowed users to distribute specific tasks to communities of paid or volunteering humans [252]. RABj provided a programmable interface for access to its human services such as entity matching. The DB-Wiki platform was designed to support collaborative data management with versioning and provenance tracking [253].

## 9.5 Comparison of Existing Approaches

This section compares how existing approaches to human-in-the-loop address the challenges of human tasks and crowdsourcing. Table 9.1 presents a summary. It should be noted that this is a high-level comparative analysis and is not exhaustive in terms of coverage of state-of-the-art literature.

**Task Specification** The complexity of task specification options available increases from augmented algorithms to platforms. Augmented algorithms tend to solve a specific problem within a dataspace with the help of human-in-the-loop techniques; therefore, human tasks are specified using general programming languages. As a result, there is limited agreement on formalisms for task specification among existing research proposals. Declarative approaches use a variety of methods to specify human tasks. The Deco [223] and CrowdDB [250] systems define query languages that extend SQL with new keywords to execute human tasks; additionally, both proposals provide details for query processors that support the execution of new keywords using human processors. By comparison, the Qurk system implemented multiple database operators over a crowdsourcing platform using user-defined functions in SQL. Platforms generally provide RESTful APIs along with their specialised task description models to allow programmable access to human intelligence services [228, 252]; other applications have also developed specialised plugins for existing applications to specify human tasks [251, 254].

**Table 9.1** Specific techniques employed by data management approaches for addressing the primary challenges of human-in-the-loop approaches

| | Augmented algorithms | Declarative programming | Platforms |
|---|---|---|---|
| Task specification | Custom functions | Query language extensions<br>User-defined functions | RESTful APIs<br>Descriptive languages |
| Interaction mechanisms | | Task templates<br>External platform | Task templates<br>Custom UI |
| Task assignment | Online optimisation | | Search & browse UI<br>Task recommendation |
| Result aggregation | Majority votingExpectation maximisation | Majority voting | |
| Incentive mechanisms | Cost Optimisation | Cost Optimisation | Leaderboards<br>Personal Scores<br>Volunteering<br>Auctions<br>Posted Prices |
| Latency issues | Retainer pools<br>Straggler mitigation<br>Pool maintenance<br>Active learning | Tasks batching<br>Response limits<br>Dynamic programming | |

**Interaction Mechanisms** Augmented algorithms either delegate interaction mechanisms, with human computers, to external platforms [248] or design custom user interfaces for each human task [244]. Declarative programming approaches use pre-defined templates to define the user interface of different tasks [249] or automatically generate elements of the user interface from generic templates [250]. Platforms provide task templates, and their associated interfaces [213, 254], or have customised the user interface of existing applications to accommodate human tasks [252].

**Task Assignment** The task assignment strategies for augmented algorithms vary a lot. Some proposals leave the assignment problem to the external platform, while other proposals actively address the assignment problem using online optimisation approaches such as primal-dual and multi-armed bandit models [230]. By comparison, declarative programming approaches for human computation do not directly address the assignment problem; instead, an external platform is used to assign tasks to human computers. Most platforms provide appropriate user interfaces to users for searching and browsing tasks themselves. These user interfaces are complemented with recommendation algorithms based on user profiles [115, 252]. For instance, RABj utilised spatial locality to recommend tasks by matching the current task with the previous history of workers [252], and [255] uses a cost-based algorithm to reduce the travel cost with dynamic task assignment in spatial crowdsourcing.

**Result Aggregation** At the basic level, the use of majority voting for aggregating results of a task from multiple human computers or crowd workers is standard practice [115, 252, 254]. More advanced algorithms employed expectation maximisation based methods to estimate the reliability of workers jointly and generate aggregated results [231]. When dealing with low-quality output, RABj employs voting, and a task escalation strategy that promotes the task to experts for review [252].

Analysis of approaches indicates that the evaluation of the algorithmic approaches is based on incremental improvement over time. On the other hand, declarative and application approaches mostly focus on describing the usefulness of human tasks while describing crowd behaviour. This underlines the lack of standard evaluation methods across approaches, even if they target similar data management problems. Understandably, all of the approaches are based on the relational data model except for Roomba, CAMEE, and RABj, which represent data in graph models [118, 228, 252].

Uncertainty reduction methods are required to improve the quality of outputs generated through human tasks. Active learning uses machine learning in combination with human tasks to generate high-quality results [246]. Providing positive or negative feedback to crowd workers can also improve their future contributions [243, 245]. Provenance information of human-generated data updates can help decide the quality of data [251]. Filtering workers based on their demographical or other characteristics can help reduce spamming [249]. Use of resolution rules for conflicting data updates generated by different workers helps in maintaining high-quality data [223]. Classifying a worker according to the system's trust in them reduces uncertainty across worker population [254].

**Incentive Mechanisms** The motivation of people to participate in human tasks directly depends on the incentive's mechanism. Both augmented algorithms and declarative programming focus on cost optimisation for human tasks based on rewards. In this regard, the existing approaches have employed a variety of dynamic pricing strategies to minimise costs or control budget [223, 249, 250]. Platforms have employed a wide variety of incentive mechanisms such as leaderboards [252], individual scores [252], posted prices [252, 254], volunteering [228, 252, 254], and auctions [222].

**Latency Issues** Latency issues are addressed through optimisation and heuristics. Latency management approaches for augmented algorithms include creating a retainer pool of pre-fetched workers from external platforms which are shown to have generated data collection results in seconds [239]. The retainer pool approach was further improved using optimisation and active learning [236] and has been applied to work with event systems [241]. Existing proposals for latency issues in declarative programming include batching a group of tasks together, which can reduce the time required to complete individual tasks [249], imposing limits on the time given to crowd workers [249], or employing dynamic programming to minimise latency [237, 238].

## 9.6 Human Task Service for Real-time Linked Dataspaces

Real-time data sources are increasingly forming a significant portion of the data generated in the world. This is in part due to increased adoption of the Internet of Things and the use of sensors for improved data collection and monitoring of smart environments which enhance different aspects of our daily activities in smart buildings, smart energy, smart cities, and others [1]. To support the interconnection of intelligent systems in the data ecosystem that surrounds a smart environment, there is a need to enable the sharing of data among intelligent systems.

### *9.6.1 Real-time Linked Dataspaces*

A data platform can provide a clear framework to support the sharing of data among a group of intelligent systems within a smart environment [1] (see Chap. 2). In this book, we advocate the use of the dataspace paradigm within the design of data platforms to enable data ecosystems for intelligent systems.

A dataspace is an emerging approach to data management that recognises that in large-scale integration scenarios, involving thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources [2]. Within dataspaces, datasets *co-exist* but are not necessarily fully integrated or homogeneous in their schematics and semantics. Instead, data is integrated on an *as-needed* basis with the labour-intensive aspects of data integration postponed until they are required. Dataspaces reduce the initial effort required to set up data integration by relying on automatic matching and mapping generation techniques. This results in a loosely integrated set of data sources. When tighter semantic integration is required, it can be achieved in an incremental *pay-as-you-go* fashion by detailed mappings among the required data sources.

We have created the Real-time Linked Dataspace (RLD) (see Chap. 4) as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data, knowledge graphs, and real-time stream and event processing capabilities to support a large-scale distributed heterogeneous collection of streams, events, and data sources [4]. Within this section, we focus on the human task support service of the RLD.

### *9.6.2 Human Task Service*

The human task service is concerned with providing humans-in-the-loop services to applications within the RLD. The service supports both virtual tasks (data management) and physical tasks (citizen serving) within the smart environment. Collaborative data management [115] within the RLD is enabled by distributing small data management tasks among willing users in the smart environment [221, 256]. The inclusion of users in the data management process not only helps in managing data but may help in building user trust and a sense of ownership of the dataspace.

Figure 9.2 shows a simple architecture for the human task service that includes a *Task Management* component which provides middleware for access to the users in the smart environment. Task management is decoupled from the data management for encapsulation. The core functions of the task management are: (1) *Task Assignment:* matching between tasks and users in the smart environment [256] based on characteristics of tasks or the specific requirements of tasks in terms of human capabilities [115, 221], and (2) *Quality assurance* to ensure truthful and correct responses of tasks.

### *9.6.3 Pay-As-You-Go Service Levels*

Dataspace support services follow a tiered approach to data management that reduces the initial cost and barriers to joining the dataspace. When tighter integration into the dataspace is required, it can be achieved incrementally by following the service tiers defined. The incremental nature of the support services is a core enabler of the pay-as-you-go paradigm in dataspaces. The functionality of the human task service follows the 5 Star pay-as-you-go model (detailed in Chap. 4) of the RLD. The human task service has the following levels of incremental support:

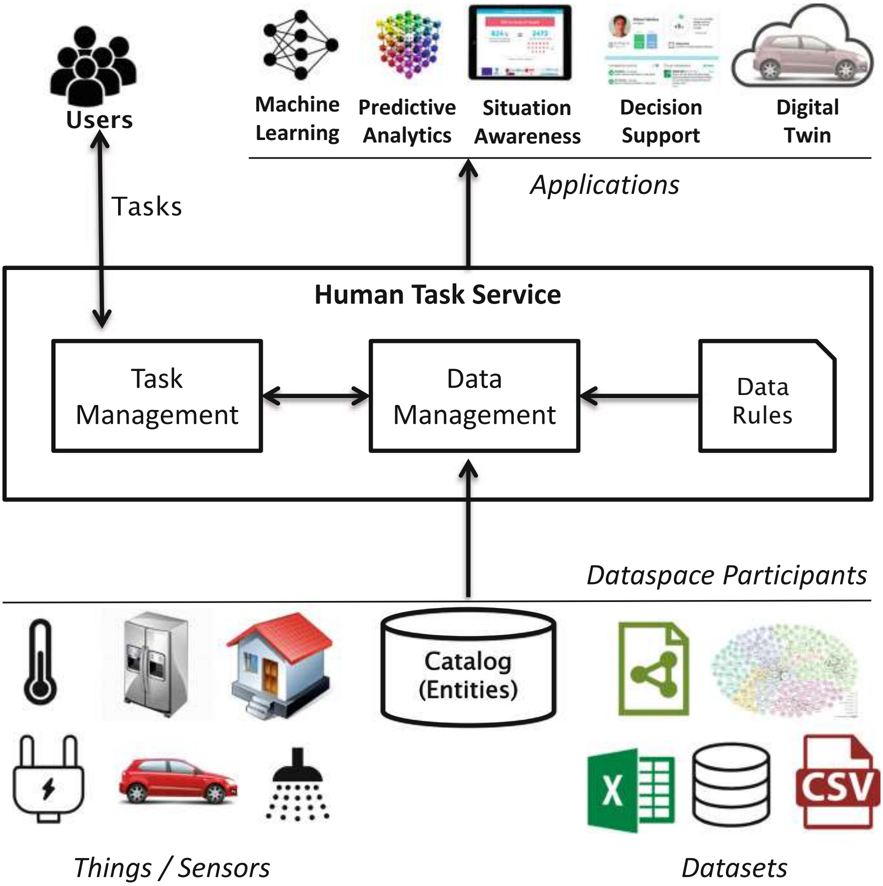
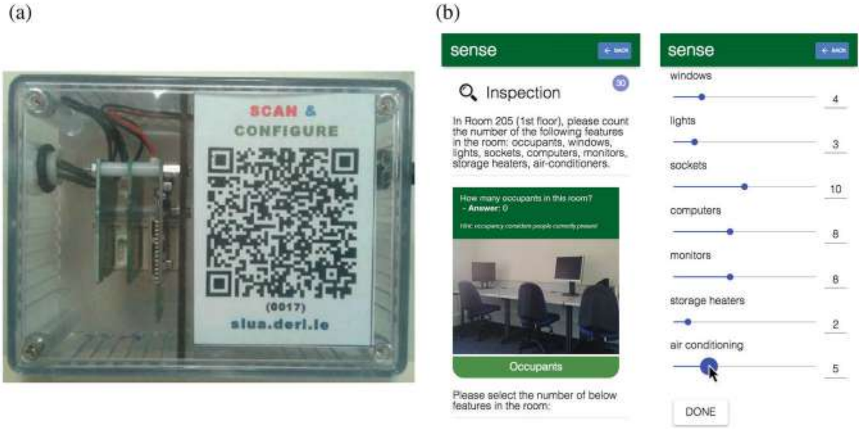


**Fig. 9.2** Overview of the human task service for Real-time Linked Dataspace [4]

*1 Star* **No Service:** No human tasks are used.
*2 Stars* **Schema:** Tasks are used to map schemas among sources.
*3 Stars* **Entity:** Tasks are used to map entities among sources.
*4 Stars* **Enrichment:** Tasks are used to enrich entities with contextual data.
*5 Stars* **Data Quality:** Tasks are used to improve the quality of data (e.g. verification).

### *9.6.4 Applications of Human Task Service*

The Human Task service may be called by applications using the RLD, or by other support services within the RLD support platform. The following four categories of human tasks are fundamental in supporting the data processing pipeline in the RLD.



In Room 205 (1st floor), please count the number of the following features in the room: occupants, windows, lights, sockets, computers, monitors, storage heaters, air-conditioners.

How many occupants in this room?
- **Answer:** 0

NB: occupancy considers people currently present

Please select the number of below features in the room:

windows 4
lights 3
sockets 10
computers 8
monitors 8
storage heaters 2
air conditioning 5

**Fig. 9.3** Examples of a human task to enrich entities. (**a**) Sensor metadata enrichment. (**b**) Entity enrichment (e.g. room features) [4]

- **Collection and Enrichment:** These types of tasks involve provisioning of data related to entities of interest in a dataspace by humans. A good illustrative example of the service in action is human tasks for entity enrichment [256]. Based on their knowledge and understanding of the environment, users can help to enrich important entities in the dataspace. Figure 9.3a shows an example enrichment task that is associated with an IoT device (e.g. CoAP sensor). The human task can be retrieved by scanning the QR code on the device with a mobile phone or tablet device. The task is retrieved from the human task service, and the user is asked a set of simple questions about the surroundings of the sensor to enrich the description of the entity in a smart building (e.g. Fig. 9.3b: what are the features of the room where the sensor is located?). Similar human tasks for data enrichment can be used across different forms of media in a dataspace, (e.g. image or video annotation).
- **Mapping and Matching:** Finding or verifying mappings among data elements of schemas and entities is another fundamental task of data integration in a dataspace that is suitable for human-in-the-loop and crowdsourcing. For example, crowdsourcing is successful in aligning ontologies in the biomedical domain [257]. A set of generalised solutions for entity resolution and matching have been proposed where they exploit human tasks to generate accurate results [258–261].
- **Operator Evaluation:** Human tasks for supporting the evaluation of database operators allow manipulation and analysis over data-in-motion and data-at-rest. This includes standard databases' operators and queries such as sort, join, and rank. For instance, human-powered evaluation of such database operators has been demonstrated for sorting [249] and skyline queries [262].
- **Feedback and Refinement:** In addition to the above human tasks, a more general set of tasks involve supporting algorithms for improvement of data quality processes, analytical models, and data transformation processes based on

feedback provided on subjective correctness or relevance of results generated by data management algorithms in the dataspace. This can range from verifying data repairs [263] to entity recognition in natural language processing [242], to providing labelled data for machine learning algorithms [264].

- **Citizen Actuation:** Moving to more physical tasks, citizen actuation tasks are formally defined as the activation of a human being as the mechanism by which a control system acts upon the environment [220]. These tasks request users to complete a physical action in the environment, such as closing a window or turning off a light. Citizen actuation tasks can take place in small-scale smart environments, like a neighbourhood/community, to medium and large-scale environments, such as a city. We envisage citizen actuation as forming part of the design process of future smart devices and environments as a method to keep users engaged with their surroundings [265].

### *9.6.5 Data Processing Pipeline*

The general pipeline for data processing in the RLD can be captured in the following four steps:

- **Data Definition:** The first step is to define the semantics and schematics of information to be processed and analysed. At the schema level, this includes the definition of concepts, entities, and their relationships, as well as specific attributes of entities. While basic semantics can be specified in the form of simple vocabularies and constraints, a more detailed semantic representation may require formal ontologies. Within the RLD, the catalog and entity management service (see Chap. 6) is used to maintain entity metadata.
- **Data Collection:** Where data acquisition is needed, the required data is collected from users by manual entry or automated tools. For example, filling out online forms, mobile applications, and entering data to a specific spreadsheet are all methods of manual data collection.
- **Data Integration and Quality:** Given that a dataspace spans multiple datasets and data sources, integration of data is a fundamental task in dataspaces that involves overcoming semantic and schematic heterogeneities of different datasets in order to present a common view of real-world entities. Data integration is performed in conjunction with data quality improvement, which can involve matching and de-duplication of schema elements and individual entities.
- **Data Analysis and Visualisation:** Users and applications pose analytical, and visualisation queries over integrated, high-quality data. Such queries are either used for creating a specific machine learning and statistical model or for serving data through appropriate graphical interfaces on different devices as required for decision-making and analysis.

Humans participate in nearly all stages of this pipeline assuming different roles and expertise such as administrators, data entry operators, integration developers, data analysts/scientists. Data administration includes but is not limited to access control, data serialisation, query management, replication, and fault tolerance. A considerable amount of research has been dedicated to the automation of data processing pipelines. However, automation solutions suffer from accuracy issues and produce uncertain results, thus requiring constant human supervision.

### *9.6.6 Task Data Model for Micro-tasks and Users*

We used the *Semantically Linked Users and Actions* (SLUA) ontology for modelling micro-tasks and users [266] within the human task service. Figure 9.4 outlines the main classes and properties in the SLUA ontology.

The main classes of the ontology are:

- **Action:** Represents a specific act that is performed by the members of the crowd. An action can be cognitive or physical. For example, the comparison of two images involves a cognitive action from the user.
- **Task:** Defines the unit of work resulting in the desired outcome that is assigned to the members of the crowd. A task may require one or more actions to produce the outcome. Therefore, a task at the lowest level is composed of actions. The Task class has a composition relationship with itself because complex tasks can be broken down into small, simple tasks.
- **User:** The class that describes the human contributor in crowdsourcing. The user serves as an intelligent agent that can perform actions for the successful completion of assigned tasks.
- **Reward:** Associated with a task as the incentive for human contribution.
- **Capability:** Defines the human characteristics that are necessary for performing a task. For instance, one system might specify a user's location capability, while another system utilises this description to assign tasks relevant to the same location.

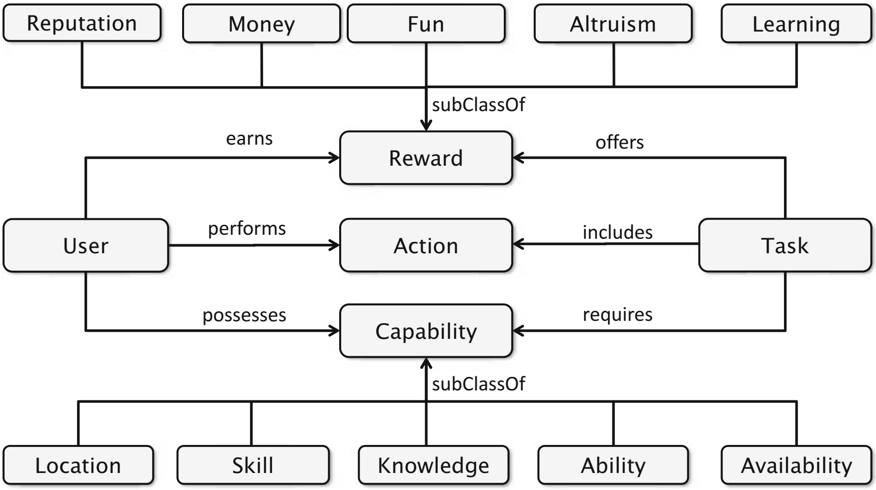


**Fig. 9.4** The SLUA ontology for micro-tasks and users

The main properties of the SLUA ontology used by the human task service are described below:

- **Domain:** A domain definition applies to most of the classes defined above. This property can be helpful for domain-specific algorithms. A common categorisation system could be used to specify domains in general crowdsourcing systems. However, for specific areas, purpose-built taxonomies can be more effective.
- **Offers:** This property defines the relationship of Reward with Task. For example, some tasks might be rewarded with money. By comparison, a user who is interested in a particular reward can be described with the "earns" property.
- **Requires:** A Task can define requirements of one or more human capabilities using this property. In contrast, a User can be described by having similar capabilities using the "possesses" property.
- **Includes:** A Task can define one or more actions that a User performs for generating the desired outcome of a task.
- **isPartOf:** A complex Task can be decomposed into small manageable tasks. Therefore, this property helps in describing the composition relationship between tasks.
- **hasDeadline:** This property can be used to specify the time limitations of a Task, which is specifically important for real-time systems employing crowds.
- **isConnectedWith:** In the context of social networks, users are connected with other users through various relations. This property captures the network structure of users to enable social network based analysis of actions and users. For example, the network structure can be exploited to recommend actions to neighbour nodes in a network.

### 9.6.7 *Spatial Task Assignment in Smart Environments*

A major challenge in spatial crowdsourcing is to assign reliable workers to nearby tasks. The goal of such a task assignment process is to maximise the task completion in the face of uncertainty. This process is further complicated when tasks arrivals are dynamic, and worker reliability is unknown. Effective assignment of tasks to appropriate users at the right time is critical to dynamic smart environments. Therefore, information about a user's location and availability are required to assign tasks related to devices and "Things" around them. There are a variety of methods for sourcing the location and availability information of users. In the human task service, we use a mixture of pull and push methods for sourcing a participant's location for making assignment decisions.

**Task Pull** The linkage between physical sensing devices and tasks is made with the help of Quick Response (QR) codes. Figure 9.5 illustrates an example of a QR code attached to a sensor. Each QR code represents an encoded URL for the sensor tasks. Resolving the URL through a browser renders the tasks associated with the sensor. The pull-based assignment is suited for situations where the information about sensors and users is not available.

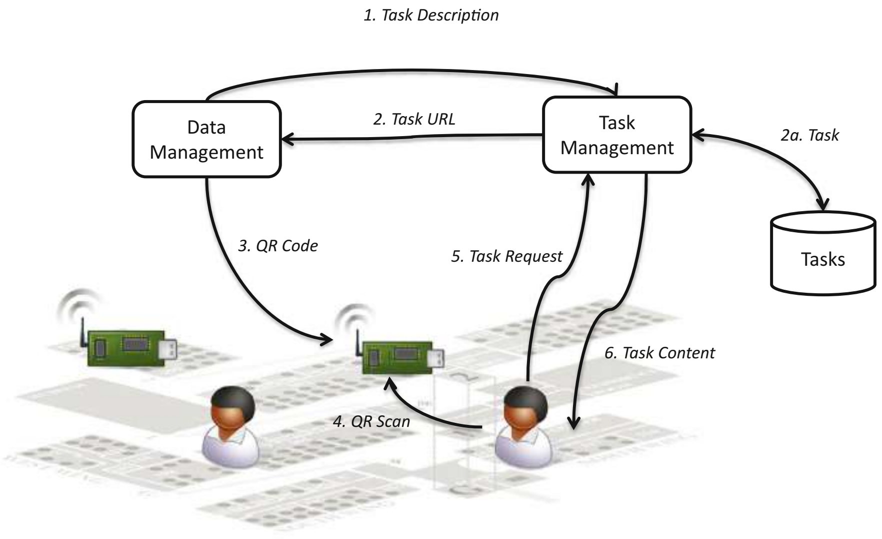


**Fig. 9.5** Example of a pull-based approach to task routing in a smart environment

**Task Push** The tasks can be actively pushed to users in situations where the location and availability information of users is available (see Fig. 9.6). To achieve this, the human task service keeps track of the sensor location information submitted by users and then pushes tasks that require the description of the surroundings of a sensor to nearby users.

Whether the pull or push method is used for assignment depends on the data management objective of the task. Therefore, the data management component must specify the assignment method for tasks, as well as the requirements in terms of human capabilities [213, 266]. Given a set of workers and their associated locations, we employ a graph-based approach to calculate their distance from the task location (see Fig. 9.7). Subsequently, the distance vector is used to optimise the assignment of a task to the appropriate worker. The optimisation objective is maximising the success rate of task completion. For this purpose, we employ an online assignment approach that aims to assign tasks to the best users while also learning about their task acceptance behaviour.

We developed a task assignment algorithm based on the multi-armed bandit model [267] as illustrated in Fig. 9.8. Our task assignment algorithm proceeds in the following manner.

- The algorithm considers the current task and the current pool of workers together with the distance vector. The vector contains the distance variables defined according to the task and worker locations.
- The algorithm chooses a worker and assigns the current task to them, based on the above information and the observed history of the previous assignments.

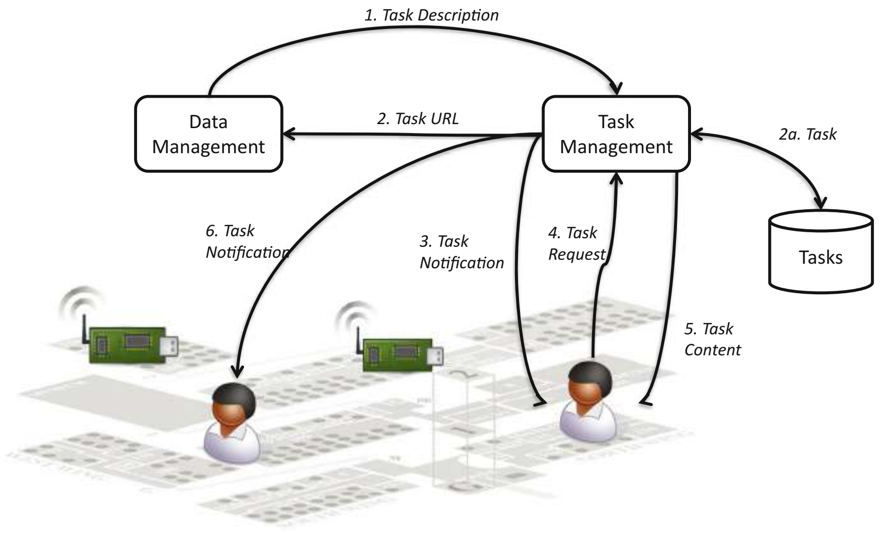


**Fig. 9.6** Example of a push-based approach to task routing in a smart environment

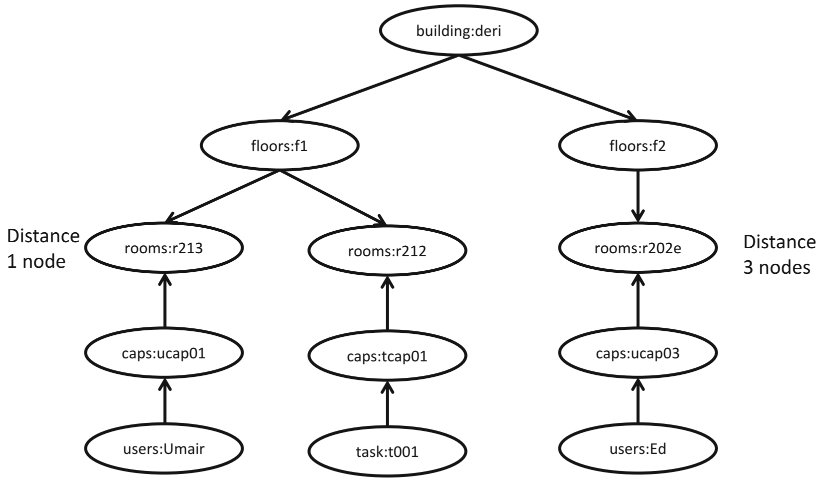


**Fig. 9.7** Hierarchical approach to a capability-based approach to matching the location of task and user

- For each assignment, the algorithm waits for the response, which depends on the current task and the chosen worker. If the worker performs the task before the expiry time, then the completed task counter is updated for the worker.

The algorithm improves its worker selection strategy based on the new observed history of the assigned tasks. The algorithm does not observe any information from the workers that are not chosen for the assignment.

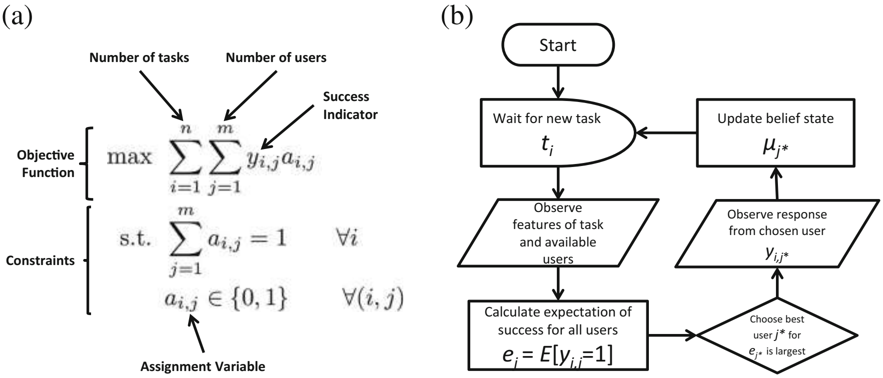

(a)
Number of tasks
Number of users
Success Indicator
Objective Function
$\max \sum_{i=1}^{n} \sum_{j=1}^{m} y_{i,j} a_{i,j}$
Constraints
$\text{s.t.} \sum_{j=1}^{m} a_{i,j} = 1 \quad \forall i$
$a_{i,j} \in \{0,1\} \quad \forall (i,j)$
Assignment Variable
(b)
Start
Wait for new task $t_i$
Observe features of task and available users
Calculate expectation of success for all users $e_j = E[y_{i,j}=1]$
Choose best user $j^*$ for $e_{j^*}$ is largest
Observe response from chosen user $y_{i,j^*}$
Update belief state $\mu_{j^*}$

**Fig. 9.8** Assignment algorithm. (**a**) Optimised task assignment using dynamic programming. (**b**) Multi-armed bandit approach for learning

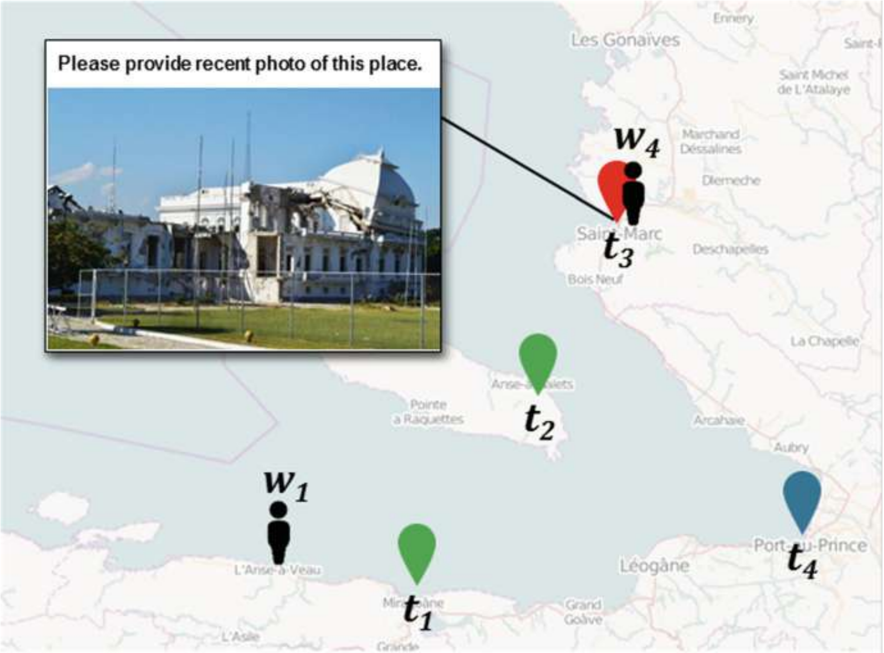


**Fig. 9.9** Example of spatial crowdsourcing on the map of Haiti after the 2010 earthquake. A new spatial task (in blue) requests recent photos of a building at the indicated location [221]

Within larger smart environments, the routing problem becomes more complex, and the assignment algorithms need to be more sophisticated. Consider the situation where a requester is interested in collecting high-quality and representative photos of disaster-hit areas in a country. The locations of interest are spread across the country. The requester designs a task for each location and is interested in the coverage of all locations with high-quality results. Figure 9.9 illustrates such a scenario on a map.

In large-scale deployments, the human task service uses a task assignment approach that combines a bi-objective optimisation with combinatorial multi-armed bandits. We formulate an online optimisation problem to maximise task reliability and minimise travel costs in spatial crowdsourcing. With the use of the *Distance-Reliability Ratio* (DRR) algorithm based on combinatorial fractional programming, we can reduce travel costs by 80% while maximising reliability when compared to existing algorithms [255]. The DRR algorithm has been extended for the scenario where worker reliabilities are unknown by using an *interval estimation* heuristic to approximate worker reliabilities. Experimental results demonstrate that the approach achieves high reliability in the face of uncertainty.

## 9.7 Summary

This chapter provides an introduction to the use of human-in-the-loop and crowdsourcing in intelligent systems within smart environments, where it can be used for virtual tasks (e.g. data management) and physical tasks (e.g. citizen actuation). The use of human-in-the-loop approaches offers exciting opportunities for utilising human processing in making sense of the data deluge and in interacting with the physical environment. However, it requires new ways of thinking about algorithms and platforms while being aware of information security, privacy, and worker exploitation issues. Within the Real-time Linked Dataspace, we have created a Human Task Service as part of the support platform. The purpose of the service is to offer human-in-the-loop support to applications within the dataspace. This chapter presented the design of the human task service and its use within human-in-the-loop applications in a smart environment, its data processing pipeline, data model, and task routing mechanisms.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

# Part III
# Stream and Event Processing Services

The third part of this book explores new techniques to support approximate and best-effort stream and event processing services for Real-time Linked Dataspaces, which support loose semantic integration and administrative proximity. This part contains chapters covering complex event processing, event service composition, stream dissemination, stream matching, and approximate semantic matching.

# Chapter 10
# Stream and Event Processing Services for Real-time Linked Dataspaces

**Keywords** Stream processing · Event processing · Real-time query · Entity-centric · Lambda architecture

## 10.1 Introduction

The goal of Real-time Linked Dataspaces is to support a real-time response from intelligent systems to situations of interest within a smart environment by providing data processing support services that follow the data management philosophy of dataspaces and meet the requirements of real-time data processing. This part of the book details support services to process streaming and event data which support loose semantic integration and administrative proximity. Support services include entity-centric queries, complex event processing, stream dissemination, and semantic matching for heterogeneous events. The goal of these services is to support a real-time linked dataspace to get setup and running with a low overhead for administrative and semantic integration costs (e.g. establishing data agreements, service selection, and service composition).

Section 10.2 of this chapter lays out pay-as-you-go services in the context of event and stream processing. Section 10.3 details the entity-centric real-time query service, including its architecture, service-levels, and service performance, and the chapter concludes with a summary in Sect. 10.4.

## 10.2 Pay-As-You-Go Services for Event and Stream Processing in Real-time Linked Dataspaces

To support the interconnection of intelligent systems in the data ecosystem that surrounds a smart environment, there is a need to enable the sharing of data among systems. A data platform can provide a clear framework to support the sharing of data among a group of intelligent systems within a smart environment [1] (see

© The Author(s) 2020
E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_10

Chap. 2). In this book, we advocate the use of the dataspace paradigm within the design of data platforms to enable data ecosystems for intelligent systems.

Dynamic data from sensors and IoT devices comprise a significant portion of data generated in a smart environment. Responding to this trend requires a data platform to provide specific support services designed to work with real-time data sources. These services must keep with the dataspace philosophy; thus, they must co-exist, and co-evolve over time, and ensure a rigid data management approach does not subsume the source systems. Within the dataspace paradigm, data management pushes the boundaries of traditional databases in two main dimensions [2]: (1) Administrative Proximity: which describes how data sources within a space of interest are close or far in terms of control, and (2) Semantic Integration: which refers to the degree to which the data schemas within the data management system are matched up.

We have created the Real-time Linked Dataspace (RLD) (see Chap. 4) as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data and real-time stream and event processing capabilities to support a large-scale distributed heterogeneous collection of streams, events, and data sources [4]. The RLD follows a set of principles to describe the specific requirements within a real-time setting:

- An RLD must deal with many different formats of streams and events.
- An RLD does not subsume the stream and event processing engines; they still provide individual access via their native interfaces.
- Queries in the RLD are provided on a best-effort and approximate basis.
- The RLD must provide pathways to improve the integration between the data sources, including streams and events, in a pay-as-you-go fashion.

In order to enable these principles to support real-time data processing for events and streams, we explore new techniques to support approximate and best-effort stream and event processing services within an RLD-Support Platform (RLD-SP). The RLD-SP services support many formats of data, do not depend on prior-agreement for composition or dissemination, and provide a best-effort quality of service and approximate answers using a pay-as-you-go approach. As shown in Fig. 10.1, the stream and event processing services provided by the RLD-SP are:

- **Entity-Centric Index:** The entity-centric index enables unified queries across live streams, historical streams, and entity data to enable full entity-centric views of the current and past state of the smart environment.
- **Stream Dissemination:** A key challenge for an RLD-SP is to disseminate events and streams to relevant data consumers efficiently. The dataspace must facilitate machine-to-machine communications to build an efficient stream dissemination system for a smart environment.
- **Complex Event Processing:** The individual and compositions of event services within a smart environment can have different quality-of-service levels (e.g. latency, accuracy, reliability). An RLD-SP must support quality-of-service aware complex event service compositions to maximise the level of service available.

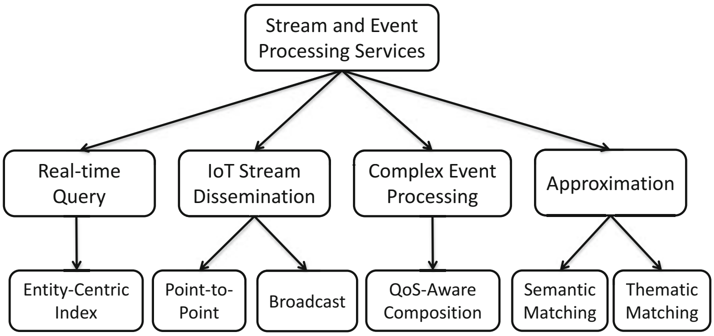


**Fig. 10.1** Support for stream and event processing in the Real-time Linked Dataspace

**Table 10.1** Pay-as-you-go stream and event support services in the RLD-SP [4]

| Pay-as-you-go star rating | Entity-centric index | Complex event processing | Stream dissemination | Semantic approximation |
|---|---|---|---|---|
| *<br>Basic | None | None | None | None |
| **<br>Machine-readable | Basic processing | Single stream | None | Semantic matching |
| ***<br>Basic integration | Historical views of streams | Multi-service composition | Point-to-point | Thematic matching |
| ****<br>Advanced integration | Stream enrichment with context and entity data | Quality of service aware service composition | Wireless broadcast | Entity-centric matching |
| *****<br>Full semantic integration, search, and query | Entity-centric real-time query | Context-aware composition | Complex patterns | Context-aware matching |

- **Approximation:** RLD-SP needs to be able to support the processing of heterogeneous events. Semantic event matchers are one approach to handle data heterogeneity within real-time events when few or no prior-agreements exist.

Each of these support services is designed following a tiered model for service provision. This means that it can provide incremental support for participants in the RLD in a "pay-as-you-go" fashion. Table 10.1 identifies possible service-tiers available from each service aligned to the RLD 5 Star pay-as-you-go model. It should be noted that not all service-tiers have been implemented within each service. Rather, the implementation of the support services also follows an incremental approach with service-tiers developed on an as-needed basis based on the actual requirements of the different smart environments. In this light, Table 10.1 also serves as a roadmap for the development of each service capturing future work for some services.

In the remainder of this part of the book, we detail the above support services and focus on their role in supporting real-time data processing in dataspaces. Each chapter details the instantiation of a support service and its evaluation. The remainder of this chapter describes the entity-centric real-time query service.

## 10.3 Entity-Centric Real-time Query Service

An essential requirement in a smart environment is the querying of real-time data streams. Within the RLD [4], this is achieved by the entity-centric real-time query service that enables unified queries across live streams, historical streams, and entity data to enable full entity-centric views of the current and past state of the smart environment. This section first discusses the Lambda Architecture and then details how it has been extended to support entity-centric real-time queries.

### *10.3.1 Lambda Architecture*

The Lambda Architecture is a frequently used Big Data processing architecture, which realises that both real-time and historical data analytics are crucial to support data analysis within smart environments. Rather than using two different systems for processing real-time data and historical data, the Lambda Architecture [268] allows seamless ingestion and processing of live and historical streaming data within a single architecture, as illustrated in Fig. 10.2.

Streams of events can be sourced from a variety of systems such as sensors, database logs, and website logs. All data entering the system is processed by both the batch layer and the speed layer. The batch layer pre-computes batch views of the stored raw data. The serving layer indexes the batch views for low-latency fast-access queries by applications. The speed layer deals with high-velocity updates by providing real-time append-only views of recent data. Queries are answered by merging results from both batch views (data-at-rest) and real-time views (data-in-motion). The Lambda Architecture has proved very useful for data management within smart environments [33].

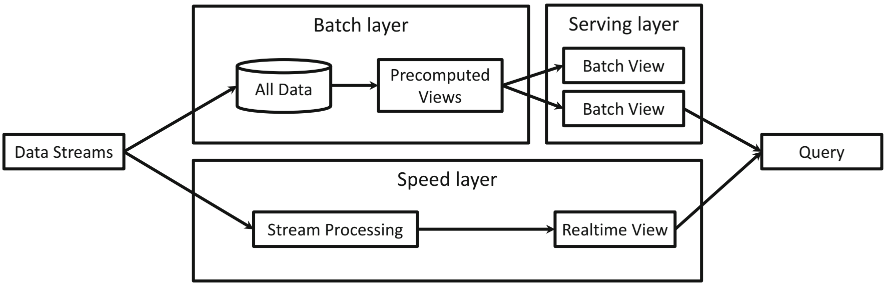


**Fig. 10.2** The three layers of the Lambda Architecture

### 10.3.2 Entity-Centric Real-time Query Service

In terms of real-time data processing, the Lambda approach meets many of the data management requirements for intelligent system data ecosystems defined in Chap. 2. However, Lambda does not natively support the inclusion of entity and contextual data within the indexing and querying process. This means that applications need to maintain the relationship between the Lambda index and the entities in the dataspace by themselves. Ideally, an entity-centric real-time query service would be provided by the RLD-SP to remove the need for applications to manage the entity/stream relationship.

To meet this requirement, we designed an extension of the Lambda Architecture that includes the addition of an entity layer for the indexing of entity data alongside historical and live streams. The approach enables the serving layer to provide merged views across all three layers, removing the need for applications to maintain the entity/stream relationships. The entity-centric real-time query service is part of the RLD-SP and is tightly integrated with other support services such as the catalog, entity management, and access control services.

Figure 10.3 illustrates the design of the entity-centric real-time query service. The main components are:

- **Entity Data (Catalog):** Provides entity data from the catalog and entity management services.
- **Data Streams:** Produced by the "things" and sensors within the smart environment.
- **Batch Layer:** Provides batch-based processing for accurate, but delayed views of historical data.
- **Speed Layer:** Provides real-time processing for data with low-latency processing requirements. Streams in the speed layer are not stored but processed on-the-fly to guarantee low-latency approximate views of the data to complement the older

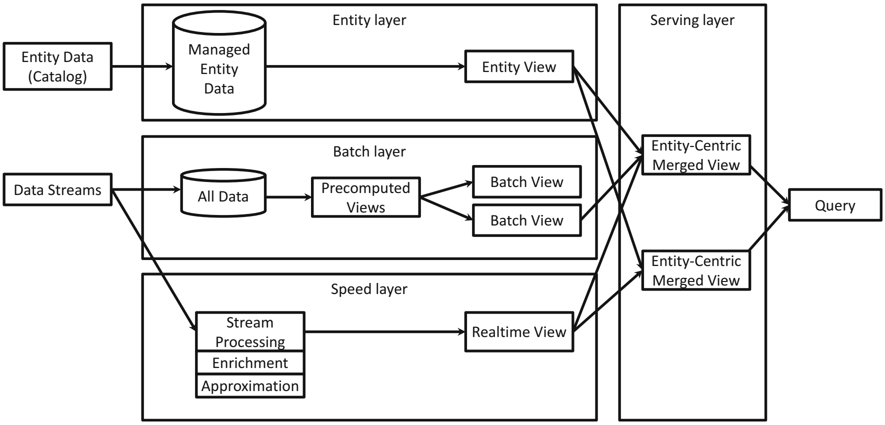


**Fig. 10.3** The four layers of the entity-centric real-time query service [4]

views achieved by the batch layer. The speed layer provides several support services for processing event data, such as approximate event matching and event enrichment.

- **Entity Layer:** Provides a view of the managed entities within the RLD working closely with the catalog and entity management service.
- **Serving Layer:** Provides applications and users with a single entity-centric query interface for data access and query. This layer transparently splits queries to the batch, speed, and entity layers, and combines pre-computed views over the three layers.
- **Query:** Request for entity-centric views from applications, analytics, and users.

### 10.3.3 Pay-As-You-Go Service Levels

Dataspace support services follow a tiered approach to data management that reduces the initial cost and barriers to joining the dataspace. When tighter integration into the dataspace is required, it can be achieved incrementally by following the service tiers defined. The incremental nature of the support services is a core enabler of the pay-as-you-go paradigm in dataspaces. The tiers of service provision provided by the entity-centric real-time query service in the RLD follows the 5 Star pay-as-you-go model (detailed in Chap. 4). The service has the following tiered-levels of support:

*1 Star* **No Service:** Streams are not managed in the service.

*2 Stars* **Basic Processing:** Basic real-time stream processing in the speed layer only.

*3 Stars* **Historical Views:** Streams are stored in the batch layer for historical views.

*4 Stars* **Enrichment:** Streams are enriched with context and entity data from the catalog and entity management service.

*5 Stars* **Entity-Centric:** Streams are processed in all three layers to provide entity-centric real-time queries.

The unified entity-centric views provided by the service proved to be beneficial to developers/data scientists using the RLD; a performance assessment of the service is provided in the next section.

### 10.3.4 Service Performance

A key contribution of a real-time linked dataspace support platform is the entity-centric real-time query service for the RLD. The query latency for the service was evaluated within each environment to ensure it could support interactive user querying [269, 270]. We evaluated seven common queries within the developed applications to determine the level of query interactivity of the service. Table 10.2

**Table 10.2** Query latency (seconds) of entity-centric real-time query service [4]

| Query type | Airport A | Mixed use A | Home A | School A | Airport B | Mixed use B |
|---|---|---|---|---|---|---|
| timeBoundary | 0.266415 | 0.076204 | 0.078805 | 0.076004 | 0.080605 | 0.078605 |
| dataSourceMetadata | 0.091405 | 0.078005 | 0.080805 | 0.153609 | 0.137808 | 0.092205 |
| segmentMetadata | 0.073804 | 0.084405 | 0.074004 | 0.140008 | 0.077405 | 0.077604 |
| Search | 0.162609 | 0.142808 | 0.085805 | 0.076404 | 0.136808 | 0.204812 |
| Timeseries | 0.072404 | 0.080005 | 0.077204 | 0.072404 | 0.134008 | 0.083605 |
| groupBy | 0.073404 | 0.078205 | 0.075604 | 0.081605 | 0.078605 | 0.072204 |
| topN | 0.078405 | 0.086805 | 0.072604 | 0.073004 | 0.077804 | 0.076604 |

presents these results based on the average of five runs of each query. Most queries have an “instantaneous response” of under 0.1 seconds, and all queries are responsive under 1 second, which is needed for “good navigation” [270]. This initial evaluation demonstrates the suitability of the query service for serving intelligent applications within smart environments.

## 10.4 Summary

The chapter provides a high-level overview of the real-time data processing support services within Real-time Linked Dataspaces (RLDs). Each service follows the pay-as-you-go data management philosophy of dataspaces. The goal of the services is to support an RLD to get setup and running with a low overhead for administrative and semantic integration costs (e.g. establishing data agreements, service selection, and service composition). This requires us to explore new techniques to support approximate and best-effort stream and event processing services which support loose semantic integration and administrative proximity. Specialised support services include complex event processing, event service composition, stream dissemination, stream matching, and approximate semantic matching. The chapter details the entity-centric real-time query service, including its architecture and service performance.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

# Chapter 11
# Quality of Service-Aware Complex Event Service Composition in Real-time Linked Dataspaces

**Feng Gao and Edward Curry**

**Keywords** Complex event processing · Quality-of-service · Modelling · Service composition · Dataspaces · Internet of Things

## 11.1 Introduction

The proliferation of sensor devices and services along with the advances in event processing brings many new opportunities as well as challenges for intelligent systems. It is now possible to provide, analyse, and react upon real-time, complex events in smart environments. When existing event services do not provide such complex events directly, an event service composition may be required. However, it is difficult to determine which event service candidates (or service compositions) best suit users' and applications' quality-of-service requirements. A sub-optimal service composition may lead to inaccurate event detection and lack of system robustness. In this chapter, we address these issues by first providing a Quality-of-Service (QoS) aggregation schema for complex event service compositions, and then developing a genetic algorithm to create near-optimal event service compositions efficiently. The approach is evaluated with both real sensor data collected via Internet of Things services and synthesised datasets.

The chapter is organised as follows: Sect. 11.2 introduces the technical aspects of complex event processing within dataspaces, including the design of the service, pay-as-you-go service levels, and its life cycle. Section 11.3 presents the Quality-of-Service (QoS) model we use and the QoS aggregation rules we define. Section 11.4 presents the heuristic that enables QoS-aware event service compositions based on Genetic Algorithms. Section 11.5 evaluates the proposed approach. Section 11.6 discusses related works in QoS-aware service planning, and Sect. 11.7 concludes and details future work.

© The Author(s) 2020
E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_11

## 11.2 Complex Event Processing in Real-time Linked Dataspaces

Real-time data sources are increasingly forming a significant portion of the data generated in the world. This is in part due to increased adoption of the Internet of Things (IoT) and the use of sensors for improved data collection and monitoring of smart environments, which enhance different aspects of our daily activities in smart buildings, smart energy, smart cities, and others [1]. To support the interconnection of intelligent systems in the data ecosystem that surrounds a smart environment, there is a need to enable the sharing of data among intelligent systems.

### *11.2.1 Real-time Linked Dataspaces*

A data platform can provide a clear framework to support the sharing of data among a group of intelligent systems within a smart environment [1] (see Chap. 2). In this book, we advocate the use of the dataspace paradigm within the design of data platforms to enable data ecosystems for intelligent systems.

A dataspace is an emerging approach to data management which recognises that in large-scale integration scenarios, involving thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources [2]. Within dataspaces, datasets *co-exist* but are not necessarily fully integrated or homogeneous in their schematics and semantics. Instead, data is integrated on an *as-needed* basis with the labour-intensive aspects of data integration postponed until they are required. Dataspaces reduce the initial effort required to set up data integration by relying on automatic matching and mapping generation techniques. This results in a loosely integrated set of data sources. When tighter semantic integration is required, it can be achieved in an incremental *pay-as-you-go* fashion by detailed mappings among the required data sources.

We have created the Real-time Linked Dataspace (RLD) (see Chap. 4) as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data, knowledge graphs, and real-time stream and event processing capabilities to support a large-scale distributed heterogeneous collection of streams, events, and data sources [4]. In this chapter, we focus on the complex event processing support service of the RLD.

### *11.2.2 Complex Event Processing*

Complex Event Processing (CEP) detects composed/complex events from real-time data streams according to predefined *Event Patterns*. It is a key enabling technology for smart cities [271], due to the inherent dynamicity of data and applications.

However, within a dataspace [19], there is a multitude of heterogeneous event sources to be discovered and integrated [272]. This poses a classic source selection problem where it is crucial to determine which event services should be used and how to compose them to match non-functional requirements defined by users or applications [273]. The problem of source selection for dataspaces is also tackled in Chap. 15.

Consider an intelligent travel-planning system using traffic sensors deployed in a city, for example, traffic sensors in the city of Aarhus, Denmark, as shown in Fig. 11.1. The events produced by the sensors are all made available in a dataspace for smart city data. A user (say "Alice") might need to plan her trip based on the current traffic condition and would like to keep monitoring the traffic during her travel. Her request may form an event request with an event pattern shown as an *Event Syntax Tree (EST)* in Fig. 11.2. Another user (say "Bob") might need to deploy a long-term event request monitoring the traffic condition in his neighbourhood, and thus form a similar event request. In both cases, multiple sensors are involved, their observations are aggregated to produce complex events with coarse-grained information, and the users may have non-functional requirements for those events, for example, having an accuracy above some threshold or a latency below a specific value. Moreover, one complex event might be useful for different event requests, that is, complex events are reusable. Thus, addressing the users' functional and non-functional requirements efficiently and effectively in this context needs to

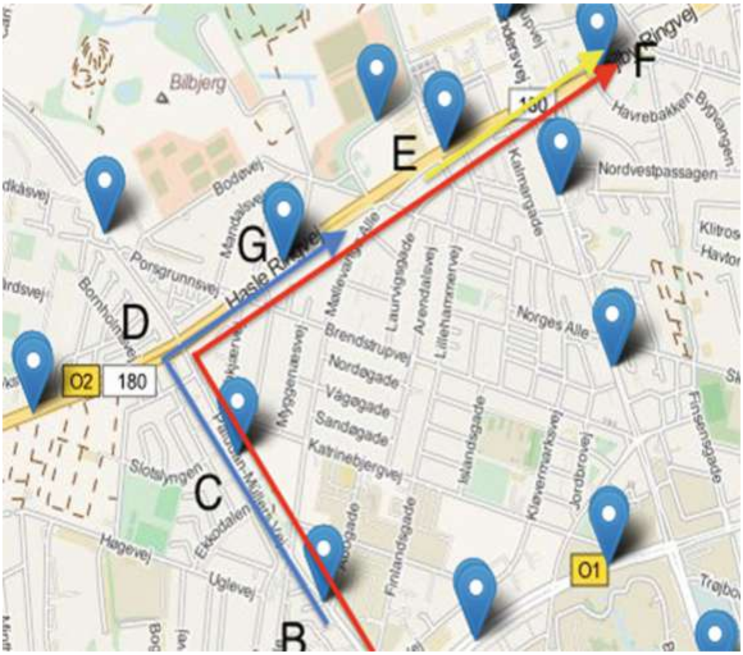


**Fig. 11.1** Traffic sensors in Aarhus City

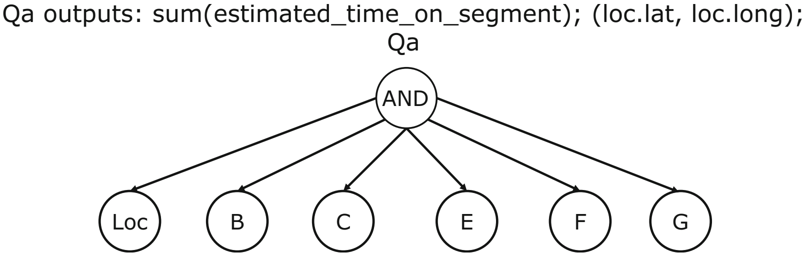

Qa outputs: sum(estimated_time_on_segment); (loc.lat, loc.long);
Qa
AND
Loc
B
C
E
F
G

**Fig. 11.2** Traffic planning event request for Alice [274]

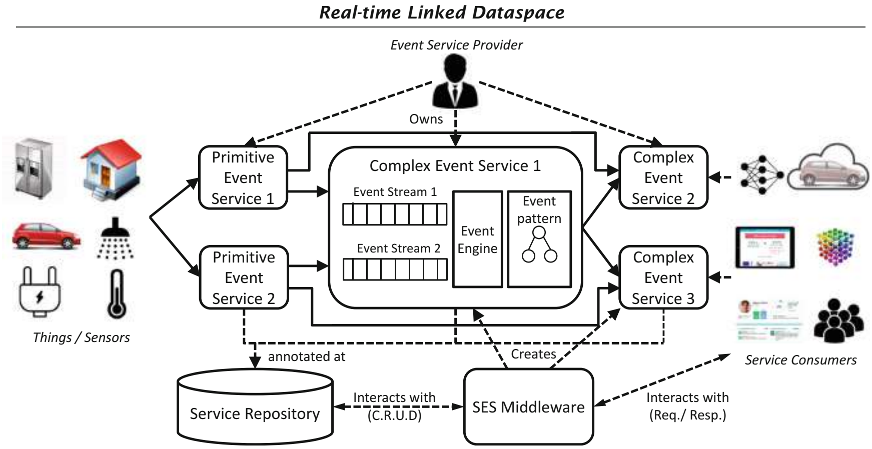


**Fig. 11.3** Overview of an event service network [276]

consider the combinations of different event sources in the dataspace as well as reusing events on different abstraction levels.

## *11.2.3 CEP Service Design*

The design of the CEP Service within the RLD [4] is based on our existing work in this area, including [123, 274–276], which is brought together in this chapter. In [275], CEP applications are provided as reusable services called Complex Event Services (CESs), and the reusability of those event services is determined by examining event patterns. Event services can thus collaborate in a distributed, cross-platform environment, creating an *Event Service Network* as shown in Fig. 11.3.

Within the RLD the complex event service composition problem is supported by the use of a specialised service to aid in event service composition.

### 11.2.4 Pay-As-You-Go Service Levels

Dataspace support services follow a tiered approach to data management that reduces the initial cost and barriers to joining the dataspace. When tighter integration into the dataspace is required, it can be achieved incrementally by following the service tiers defined. The incremental nature of the support services is a core enabler of the pay-as-you-go paradigm in dataspaces. The functionality of the complex event processing service follows the 5 Star pay-as-you-go model (detailed in Chap. 4) of the RLD. The complex event processing service has the following tiered-levels of support:

*1 Star* **No service:** No complex event processing is supported.
*2 Stars* **Single-service:** Event patterns are identified within a single stream.
*3 Stars* **Multi-service:** Event patterns can be composed of multiple event services.
*4 Stars* **QoS-aware:** Quality-of-Service (QoS) aware service composition of event services.
*5 Stars* **Context-aware:** Context-aware event processing with the use of knowledge from the dataspace.

In this chapter, we detail the implementation of the CES for the RLD with the aim to enable a QoS-aware event service composition and optimisation.

### 11.2.5 Event Service Life Cycle

In order to understand the problem of realising event service composition and optimisation, we analyse the different activities related to event services from their creation to termination. We identify the following five key activities in the life cycle of event services, as depicted in Fig. 11.4:

0. *Service Description:* The static description of the service metadata is created and stored in the service repository. Describing services and storing the descriptions is a preliminary step for any service requests to be realised by the described services.
1. *Request Definition:* An event service consumer identifies the requirements on the interested complex events (as well as the services that deliver the events) and specifies those requirements in an event service request.
2. *Planning:* An agent receives a consumer's request and matches it against service descriptions in the service repository. If direct matches are found, the matching service descriptions are retrieved, and the matching process ends. Otherwise,

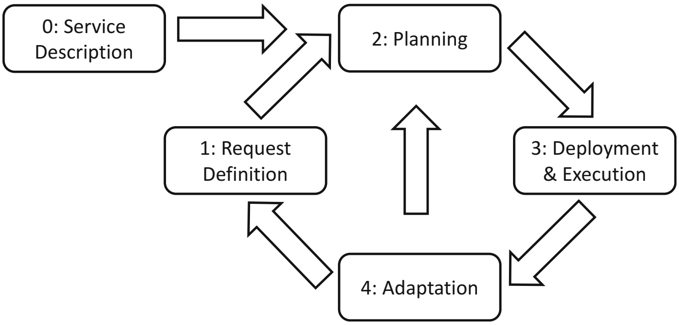


**Fig. 11.4** Life cycle of an event service [276]

existing event services are composed to fulfil the requirements, and composition plans are derived.

3. *Deployment and Execution:* An agent establishes connections between the event service consumer and providers by subscribing to event services (for the consumer) based on a composition plan, then it starts the event detection (if necessary) and messaging process.
4. *Adaptation:* An agent monitors the status of the service execution to find irregular states. When irregular states are detected, the planning activity is invoked to create new composition plans and/or service subscriptions. If the irregular states occur too often, it may suggest that the service request needs to be re-designed.

We consider efficient and effective management of the event service life cycle having the following three basic requirements:

- *User-Centric Event Request Definition:* The event requests should reflect each individual user's requirements or constraints on both Functional Properties (FP) and Non-Functional Properties (NFP) of complex events. Users should be able to specify different events they are interested in by specifying FP, for example, event type and pattern. Additional to FP, it is very likely that different users may have different sets of preferences for the NFPs: some may ask for accurate results while others may ask for more timely notifications. The implemented event services should be capable of tackling these requirements and constraints.
- *Automatic Event Service Planning:* The service planning activity should be able to automatically discover and compose CESs according to users' functional and non-functional requirements. Planning based on the functional aspects requires comparing the semantic equivalence of event patterns, while planning based on the non-functional aspects requires calculating and comparing the composition plans with regard to the QoS parameters. To fully benefit from automatic implementation and enable an on-demand event service implementation, the automatic planning should be efficient to be carried out at run-time.

- *Automatic Event Service Implementation:* The deployment of the composition plans should also be automatic to facilitate automatic execution. The adaptation activity should have the ability to automatically detect service failures or constraint violations according to users' requirements at run-time and make appropriate adjustments, including re-compose and re-deploy composition plans, to adapt to changes. The adaptation process should be efficient to minimise information loss and maximise the performance of the event services over time.

Within the RLD, the complex event service composition problem is supported by the use of a specialised service to aid in event service composition. Two issues should be considered in the design of the CES: QoS aggregation and composition efficiency. The QoS aggregation for a complex event service relies on how its member event services are correlated and composed. The aggregation rules are inherently different from conventional web services. Efficiency becomes an issue when the complex event consists of many primitive events, and each primitive event detection task can be achieved by multiple event services. We address both issues by (1) creating QoS aggregation rules and utility functions to estimate and assess QoS for event service compositions, and (2) enabling efficient event service compositions and optimisation regarding QoS constraints and preferences based on Genetic Algorithms.

## 11.3 QoS Model and Aggregation Schema

To have a comprehensive approach for QoS-aware event service composition and optimisation within a dataspace, we need an objective function. In this section, we first discuss the relevant QoS dimensions for event services. Then, we briefly explain how we aggregate them in an event service composition and how we derive the event QoS utility as our objective function.

### *11.3.1 QoS Properties of Event Services*

For an event service (or event service composition), its overall QoS can be discussed on several dimensions. In this work, we consider QoS attributes from [65] that are relevant for QoS propagation and aggregation, including:

- *Latency (L):* Describes the delay in time for an event transmitted by the service*Price (P):* Describes the monetary costs for an event service.
- *Energy Consumption (Eng):* Describes the energy costs for an event service.
- *Network Consumption (Net):* Describes the usage of a network of an event service, measured by messages consumed per unit time.
- *Availability (Ava):* Describes the possibility (in percentages) of an event service being accessible.

- *Completeness (C):* Describes the completeness (in percentages) of events delivered by an event service.
- *Accuracy (Acc):* Describes the possibility (in percentages) of getting correct event messages.
- *Security (S):* Describes the security levels (higher numerical value indicates higher security levels).

By the above definition, a quality vector $Q = <L, P, Eng, Net, Ava, C, Acc, S>$ can be specified to indicate the QoS performance of an event service in eight dimensions.

### 11.3.2 QoS Aggregation and Utility Function

The QoS performance of an event service composition is influenced by three factors: *Service Infrastructure*, *Composition Pattern*, and *Event Engine*. The *Service Infrastructure* refers to computational hardware, service Input/Output (I/O) implementation, and the physical network connection; it determines the inherent I/O performance of a service. The *Composition Pattern* refers to the way that the member event services are correlated, expressed in event patterns. The internal implementation of the *Event Engine* also has an impact on the QoS. Table 11.1 summarises how the different QoS parameters of an event service composition are calculated based on these three factors. In this chapter, we do not elaborate on the impact of service infrastructure or event engine but focus on the QoS aggregation over the composition pattern, that is, how different QoS dimensions propagate over different event service correlations, which is summarised in Table 11.2. We apply

**Table 11.1** Overall Quality of Service calculation [274]

| | QoS symbols | | | |
|---|---|---|---|---|
| Dimensions | Service infrastructure | Composition pattern | Event engine | Overall QoS calculation |
| Latency | $L_i$ | $L_c$ | $L_e$ | $L = L_i + L_c + L_e$ |
| Price | $P_i$ | $P_c$ | – | $P = P_i + P_c$ |
| Energy | $Eng_i$ | $Eng_c$ | $Eng_e$ | $Eng = Eng_i + Eng_c + Eng_e$ |
| Network consumption | – | $Net_c$ | – | $Net = Net_c$ |
| Availability | $Ava_i$ | $Ava_c$ | – | $Ava = Ava_i \times Ava_c$ |
| Completeness | $C_i$ | $C_c$ | $C_e$ | $C = C_i \times C_c \times C_e$ |
| Accuracy | $Acc_i$ | $Acc_c$ | $Acc_e$ | $Acc = Acc_i \times Acc_c \times Acc_e$ |
| Security | $S_i$ | $S_c$ | – | $S = \min(S_i, S_c)$ |

**Table 11.2** Quality of Service aggregation rules based on composition patterns [274]

| QoS dimensions for event service $\mathscr{E}$ | Aggregation rules | Applicable event operators |
|---|---|---|
| $\mathrm{P}_c(\mathscr{E})$ | $\sum_{e \in \mathcal{E}_{\mathrm{ice}}} \mathrm{P}_c(e)$ | And, Or, Sequence, Repetition |
| $\mathrm{Eng}_c(\mathscr{E})$ | $\sum_{e \in \mathcal{E}_{\mathrm{ice}}} \mathrm{Eng}_c(e)$ | And, Or, Sequence, Repetition |
| $\mathrm{Net}_c(\mathscr{E})$ | $\sum_{e \in \mathcal{E}_{\mathrm{ice}}} \mathrm{C}_c(e) \cdot \mathrm{f}(e)$ | And, Or, Sequence, Repetition |
| $\mathrm{Ava}_c(\mathscr{E})$ | $\prod_{e \in \mathcal{E}_{\mathrm{ice}}} \mathrm{Ava}_c(e)$ | And, Or, Sequence, Repetition |
| $\mathrm{Acc}_c(\mathscr{E})$ | $\prod_{e \in \mathcal{E}_{\mathrm{ice}}} \mathrm{Acc}_c(e)$ | And, Or, Sequence, Repetition |
| $\mathrm{S}_c(\mathscr{E})$ | $\min \{\mathrm{S}_c(e) \mid e \in \mathcal{E}_{\mathrm{ice}}\}$ | And, Or, Sequence, Repetition |
| $\mathrm{L}_c(\mathscr{E})$ | $\mathrm{L}_c(e), e$ is the last event in $\mathcal{E}_{\mathrm{dse}}$ | Sequence, Repetition |
| | $\mathrm{avg}\ \{\mathrm{L}_c(e) \mid e \in \mathcal{E}_{\mathrm{dse}}\}$ | And, Or |
| $\mathrm{C}_c(\mathscr{E})$ | $\frac{\min \{\mathrm{C}_c(e) \cdot \mathrm{f}(e) \mid e \in \mathcal{E}_{\mathrm{dse}}\}}{\mathrm{card}\ (\mathcal{E}) \cdot \mathrm{f}(\mathcal{E})}$ | And, Sequence, Repetition |
| | $\frac{\max \{\mathrm{C}_c(e) \cdot \mathrm{f}(e) \mid e \in \mathcal{E}_{\mathrm{dse}}\}}{\mathrm{card}\ (\mathcal{E}) \cdot \mathrm{f}(\mathcal{E})}$ | Or |

the rules in Table 11.2 from leaves to the root of a composition pattern to derive the overall QoS step-by-step. We refer readers to [275] for a more thorough explanation of Tables 11.1 and 11.2.

### 11.3.3 Event QoS Utility Function

Given a quality vector of an event service composition Q = <L, P, Eng, Net, Ava, C, Acc, S > representing the service QoS capability, we denote q as one of the eight quality dimensions in the vector, O($q$) as the theoretical optimum value (e.g. for latency the optimum value is 0 s) in the quality dimension of q, C($q$) as the user-defined value specifying the hard constraints (i.e. worst acceptable value, e.g. 1 s for latency) on the dimension, and $0 \leq \mathrm{W}(q) \leq 1$ as the weighting function of the quality metric, representing users' preferences (e.g. W(L) = 1 means latency is very important for the user and W(L) = 0 means latency is irrelevant for the user). Furthermore, we distinguish between QoS properties with positive or negative tendency: $\mathrm{Q} = \mathrm{Q}_{+} \cup \mathrm{Q}_{-}$, where $\mathrm{Q}_{+} = \{\mathrm{Ava, C, Acc, S}\}$ is the set of properties with the positive tendency (larger values the better), and $\mathrm{Q}_{-} = \{\mathrm{L, P, Eng, Net}\}$ is the set of properties with the negative tendency (smaller values the better). The QoS utility U is derived by: $= \sum_{q_i \in Q_{+}} \frac{W(q_i) \cdot (q_i - C(q_i))}{O(q_i) - C(q_i)} - \sum_{q_j \in Q_{-}} \frac{W(q_j) \cdot (q_j - O(q_j))}{C(q_j) - O(q_{j.})}$

According to the above equation, the best event service composition should have the maximum utility U. A normalised utility with values between [0,1] can be derived using the function $\overline{U} = (U + |Q_-|)/(|Q_+| + |Q_-|)$.

## 11.4 Genetic Algorithm for QoS-Aware Event Service Composition Optimisation

Event service composition is inherently an NP-hard problem; hence, we propose a Genetic Algorithm (GA) to find a near-optimal solution in a reasonable time. Typically, a GA-based search iterates the process of population initialisation, select, crossover, and mutation to maximise the "fitness" (QoS utility introduced in Sect. 11.3.3) of the solutions in each generation.

### *11.4.1 Population Initialisation*

During population initialisation, we generate individual composition plans as Concrete Composition Plans (CCPs). CCPs are event patterns with specific event service correlations and service bindings for the implementation of event service compositions, that is, each CCP is an individual solution for the event service composition problem. CCPs are generated from Abstract Composition Plans (ACPs), which are composition plans without service bindings. ACPs come from the event service composition request; we mark the reusable nodes in the requested event pattern (as shown in Fig. 11.5) by identifying isomorphic sub-graphs (as in [275]). Then, by enumerating all combinations of the implementation of the sub-patterns with reusable nodes as roots, we can list all ACPs. Finally, we pick a random subset of ACPs and generate a set of CCPs by binding event services.

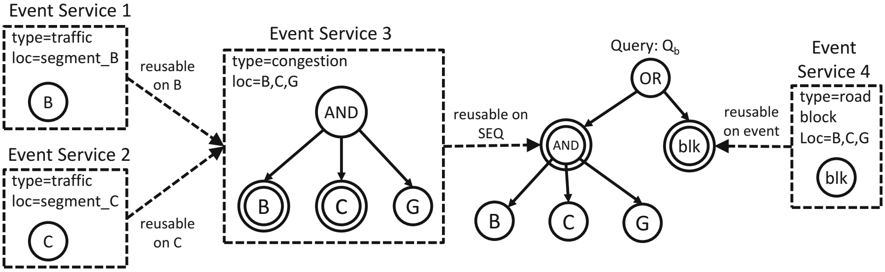


**Fig. 11.5** Marking the re-usable nodes [274]

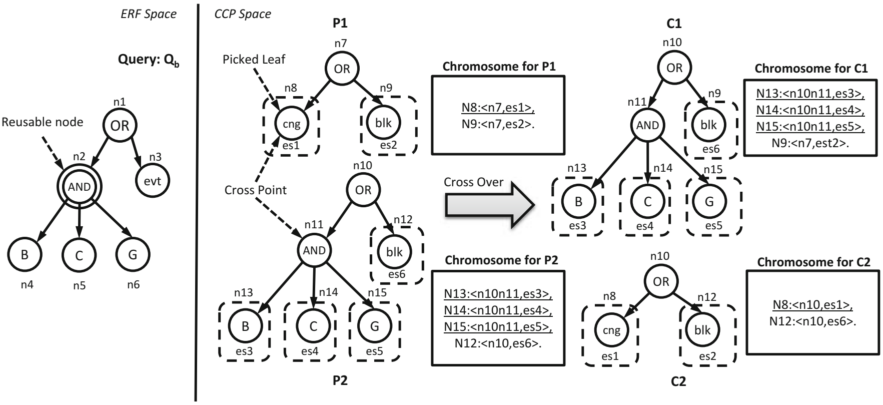


**Fig. 11.6** Example of genetic encodings and crossover [274]

### 11.4.2 Genetic Encodings for Concrete Composition Plans

Individuals (CCPs) in the population need to be genetically encoded to represent their various characteristics (composition patterns). Conventionally, web service compositions are encoded with a sequence of service identifiers, the index of each service identifier correlates it to the specific service task in the composition. We follow this principle and encode each leaf in a CCP with an event service identifier in tree traversal order. However, since the event syntax tree is partially ordered, the position or index of the event service identifier is insufficient to represent its corresponding task.

Moreover, ancestor operators of the leaf nodes can help with identifying the role of the leaf nodes in the CCPs. Therefore, global identifiers are assigned to all the nodes in the CCPs, and a leaf node in a CCP is encoded with a string of node identifiers as a prefix representing the path of its ancestors and a service identifier indicating service binding for the leaf, as shown in Fig. 11.6. For example, a gene for the leaf node "n13" in $P_2$ is encoded as a string with the prefix "n10n11" and a service ID for the traffic service candidate for road segment B, that is, "es3"; hence the full encoding of n13 is <n10n11,es3>. The complete set of encodings for every gene constitutes the chromosome of $P_2$.

### 11.4.3 Crossover and Mutation Operations

After the population initialisation and encoding, the algorithm iterates the cycle of select, crossover, and mutation to find optimal solutions. The selection is trivial; individuals with better finesses (i.e. QoS utility) are more likely to be chosen to reproduce. In the following, we explain the details on the crossover, mutation, and elitism operations designed for GA-based event service composition.

#### 11.4.3.1 Crossover

To ensure that the crossover operation produces valid child generations, parents must only exchange genes representing the same part of their functionalities, that is, the same (sub-) event detection task, specified by semantically equivalent event patterns. An example of crossover is illustrated in Fig. 11.6. Given two genetically encoded parent CCPs $P_1$ and $P_2$, the event pattern specified in the query $Q$ and the Event Reusability Hierarchy *(ERH)*,[1] the crossover algorithm takes the following steps to produce the children:

1. Pick a leaf node l1 randomly from $P_1$; create the node type prefix $ntp_1$ from the genetic encoding of $P_1$, that is, $code_1$, as follows: replace each node ID in the prefix of $code_1$ with the operator type.
2. For each leaf $l_1$ in $P_2$, create the node type prefix $ntp_2$ from $code_2$ (i.e. encodings for $l_2$) and compare it with $ntp_1$. If $ntp_1 = ntp_2$ and the event semantics of $l_1$ and $l_2$ are equivalent, that is, they are merged into the same node in the *ERH*, then mark $l_1$, $l_2$ as the crossover points $n_1$, $n_2$. If $ntp_1 = ntp_2$ but the pattern of $l_1$ is reusable to $l_2$ or $l_2$ is reusable to $l_1$, then search back on $code_1$, $code_2$ until the cross points $n_1$, $n_2$ are found on $code_1$, $code_2$ such that $T(n_1) \doteq T(n_2)$, that is, the sub-patterns of $P_1$, $P_2$ with $n_1$, $n_2$ as the root node of the *Event Syntax Tree* (EST) of the sub-patterns are semantically equivalent.
3. If $ntp_1$ is an extension of $ntp_2$, for example, $ntp_1 = (And;Or;Seq)$, $ntp_2 = (And;Or)$ and the pattern of $l_1$ is reusable to $l_2$ in the *ERH*, then search back on $code_1$ and try to find $n_1$ such that the sub-pattern with EST $T(n_1)$ is equivalent to $l_2$. If such $n_1$ is found, mark $l_2$ as $n_2$.
4. If $ntp_2$ is an extension of $ntp_1$, do the same as step 3 and try to find the cross point $n_2$ in $code_2$.
5. Whenever the cross points $n_1$, $n_2$ are marked in the previous steps, stop the iteration. If $n_1$ or $n_2$ is the root node, return $P_1$, $P_2$ as they were. Otherwise, swap the sub-trees in $P_1$, $P_2$ whose roots are $n_1$, $n_2$ (and therefore the relevant genes), resulting in two new CCPs.

#### 11.4.3.2 Mutation and Elitism

We apply a *Mutation* operation (with a certain possibility called mutation rate) after each crossover. The mutation operation randomly changes the composition plan for a leaf node in a CCP. The result of the mutation could be a different service binding for the leaf or replacing the leaf node with a new composition using the leaf node as an event request.

[1] An ERH is a DAG with nodes representing event patterns and edges representing the reusable relations, we introduce the ERH in our previous work in [123].

We apply an *Elitism* operation after each selection of a generation and add an exact copy of the best individual from the previous generation. Elitism allows us to ensure the best individual will survive over multiple generations.

## 11.5 Evaluation

In this section, we present the evaluation results of the proposed approaches. We put our experiments in the context of an intelligent travel-planning system using both real and synthetic sensor datasets for the city of Aarhus. In this scenario, a user will select a travel route and make an event request which tries to continuously monitor the traffic condition using the sensors deployed along the route that are available in the dataspace. The evaluation has two parts: in the first part, we analyse in detail the performance of the GA. In the second part, we demonstrate the usefulness of the QoS aggregation rules. All experiments are carried out on a machine with a 2.53 GHz duo core CPU and 4 GB 1067 MHz memory. Experiment results are an average of 30 iterations.

### *11.5.1 Part 1: Performance of the Genetic Algorithm*

In this part of the evaluation, we compare the QoS utility derived by Brute-Force (BF) enumeration and the developed GA. Then, we test the scalability of the GA. Finally, we analyse the impact of different GA parameters and provide guidelines to identify optimal GA parameter settings.

#### 11.5.1.1 Datasets

Open Data Aarhus (ODAA) is a public platform that publishes sensor data and metadata about the city of Aarhus. Currently, there are 449 pairs of traffic sensors in ODAA. Each pair is deployed on one street segment for one direction and reports the traffic conditions on the street segment. These traffic sensors are used in the experiments to answer requests on travel planning. We also include some other sensors in our dataset that might be used in traffic monitoring and travel planning, for example, air pollution sensors and weather sensors. These sensors are not relevant to requests like Alice's (i.e. denoted Qa in Fig. 11.2) or Bob's (denoted Qb), that is, they are noise to queries like Qa and Qb (but could be used in other travel-related queries). In total, we use 900 real sensors from ODAA, in which about half of them are noise. We denote this dataset sensor repository R0.

Each sensor in R0 is annotated with a simulated random quality vector <L, Acc, C, S>, where L $\in$ [0 ms, 300 ms], Acc, C $\in$ [50%, 100%], S $\in$ [1, 5], and frequency f $\in$ [0.2 Hz, 1 Hz]. We do not model price or energy consumption in the

**Table 11.3** Simulated sensor repositories [274]

| | $R_1$ | $R_2$ | $R_3$ | $R_4$ | $R_5$ | $R_6$ | $R_7$ | $R_8$ | $R_9$ |
|---|---|---|---|---|---|---|---|---|---|
| N | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Total size | 1800 | 2700 | 3600 | 4500 | 5400 | 6300 | 7200 | 8100 | 9000 |

**Table 11.4** Queries used in experiments [274]

| Query | Description | Nodes |
|---|---|---|
| $Q_a$ | Alice's query on estimated travel time on the route | 1 AND, 6 streams |
| $Q_b$ | Bob's query on traffic condition | 1 AND, 1 OR, 4 streams |
| $Q'_a$ | A variant of $Q_a$ with more nodes | 1 AND, 3 random operators, 8 streams |
| $Q'_b$ | A variant of $Q_b$ with more nodes | 1 AND, 1 OR, 10 streams |

experiments because their aggregation rules are similar to network consumption. For similar reasons, we also do not model availability. To test the algorithms on a larger scale, we further increase the size of the sensor repository by adding N functionally equivalent sensors to each sensor in R0 with a random quality vector, resulting in the nine different repositories as shown in Table 11.3. In the experiments, we use a loose constraint to enlarge the search space, and we set all QoS weights to 1.0. The queries used in the experiments are summarised in Table 11.4.

#### 11.5.1.2 QoS Utility Results and Scalability

In this set of experiments, we first demonstrate the usefulness of the GA by comparing it to a BF algorithm and a random pick approach. Figure 11.7 shows the experimental results for composing $Q_a$ over $R_3$ to $R_9$ ($R_1$ and $R_2$ are not tested here because their solution spaces are too small for GA), where $Q_a$ has six service nodes and one operator. A more complicated variant of $Q_a$ with eight service nodes and four operators is also tested, denoted $Q'_a$.

The best utility obtained by the GA is the highest utility of the individual in the last generation before the GA stops. In the current implementation, the GA is stopped when the current population size is less than five or the difference between the best and the average utility in the generation is less than 0.01, that is, the evolution has converged. Given the best utility from BF $\overline{U}_{bf}$, best utility from GA $\overline{U}_{ga}$, and the random utility of the dataset $\overline{U}_{rand}\}$, we calculate the degree of optimisation as $d = (\overline{U}_{ga} - \overline{U}_{rand})/(\overline{U}_{bf} - \overline{U}_{rand})$. From the results in Fig. 11.7, we can see that the average is $d = 89.35\%$ for $Q_a$ and $Q'_a$. In some cases, the BF algorithm fails to complete, for example, $Q_a$ over $R_8$ and $R_9$, because of memory limits (heap size set to 1024 MB). We can see that for smaller repositories, d is more significant. This is because, under the same GA settings (initial population size: 200, crossover rate: 95%, mutation rate: 3%), the GA has a higher chance of finding the global optimum during the evolution when the solution space is small, and the

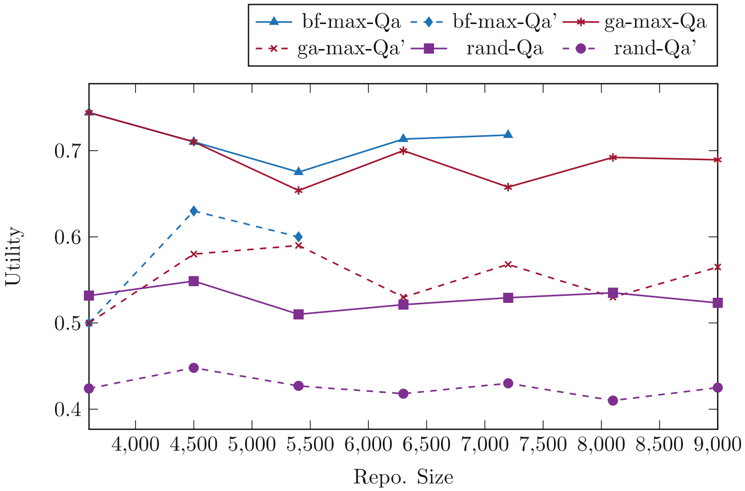


**Fig. 11.7** QoS utilities of BF, GA, and random pick [274]

*elitism* method described in Sect. 11.4.3 makes sure that, if found, the global optimum “survives” till the end of evolution, for example, in the GA results for $Q_a$ over $R_3$ and $R_4$ in Fig. 11.7.

It is evident that a BF approach for QoS optimisation is not scalable because of the NP-hard nature of the problem. We analyse the scalability of the GA using different repository sizes, query sizes (total number of event operator nodes and event service nodes in the query), as well as different number of CESs in the Event Reusability Hierarchy (ERH).

From the results in Fig. 11.8a, we can see that the composition time of $Q_a$ grows linearly for GA when the size of the repository increases. To test the GA performance with different query sizes using different operators, we use the EST of $Q_b$ as a base and replace its leaf nodes with randomly created sub-trees (invalid ESTs excluded). Then we test the GA convergence time of these queries over $R_5$. Results from Fig. 11.8b indicate that the GA execution time increases linearly regarding the query size.

In order to test the scalability over a different number of CESs in the ERH (called ERH size), we deploy 10–100 random Complex Event Services (CESs) to $R_5$, resulting in ten new repositories. We test the GA on a query created in the previous step (denoted $Q'_b$) with the size of 12 nodes (two operators, ten sensor services) and record the execution time in Fig. 11.8c. To ensure each CES could be used in the composition plan, all CESs added are sub-patterns of $Q'_b$. From the results, we can see that although the increment of the average execution time is generally linear, in some rare test instances there are “spikes”, such as the maximum execution time for ERHs of size 40 and 80. After analysing the results of those cases, we found that most (over 90%) of the time is spent on population initialisation, and the complexity of the ERH causes this, that is, the number of edges considered during ACP creation.

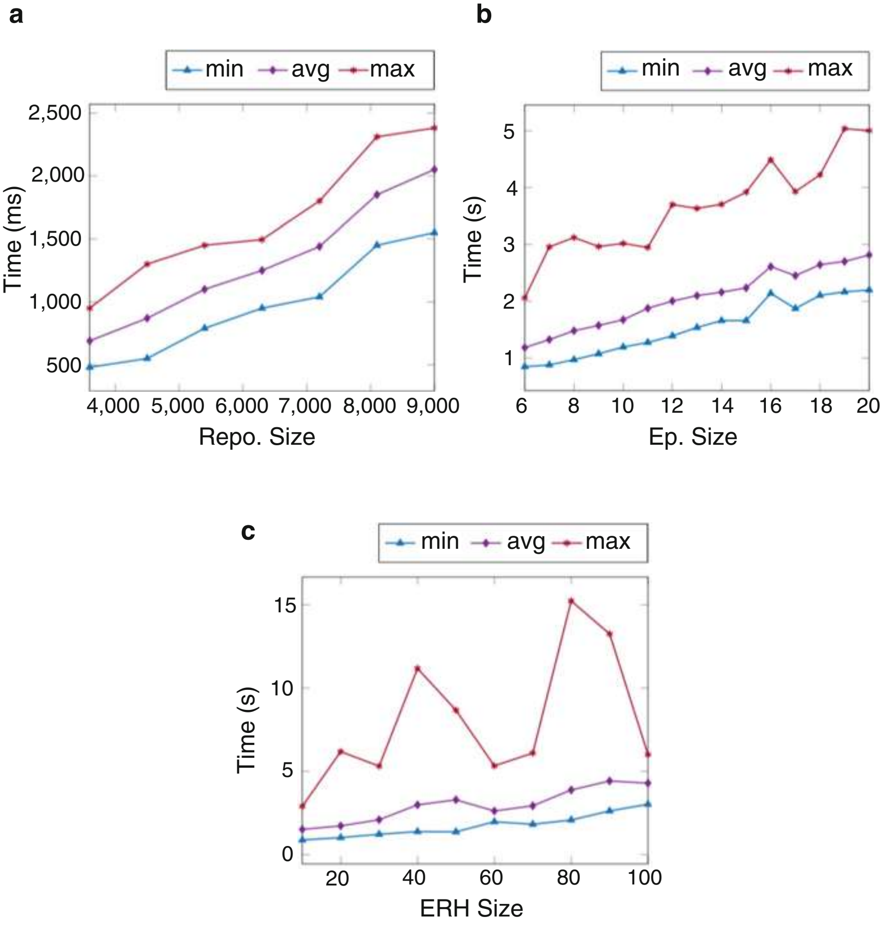


**Fig. 11.8** (**a**) GA scalability over repository size, (**b**) GA scalability over EP size, (**c**) GA scalability over ERH size [274]

### 11.5.1.3 Fine-Tuning the Parameters

In the experiments above, a fixed set of settings is used as the GA parameters, including crossover rate, mutation rate, and population size. To find good settings of the GA in our given problem domain, we fine-tune the mutation rate, population size, and crossover rate based on the default setting used above. We change one parameter value at a time while keeping other parameters unchanged.

In order to determine the effect of the parameter tuning, we define a Cost-Effectiveness score (i.e. CE-score) as follows: given the random pick utility of a dataset $\overline{\mathrm{U}}_{\mathrm{rand}}$, we have the final utility derived by GA $\overline{\mathrm{U}}_{\mathrm{ga}}$ and the number of milliseconds taken for the GA to converge $t_{\mathrm{ga}}$, CE-score = $(\overline{\mathrm{U}}_{\mathrm{ga}} - \overline{\mathrm{U}}_{\mathrm{rand}}) \times 10^5/t_{\mathrm{ga}}$. We test two queries $\mathrm{Q}_a$, $\mathrm{Q}'_b$ over two new repositories $\mathrm{R}'_5$, $\mathrm{R}'_9$, which are $\mathrm{R}_5$ and $\mathrm{R}_9$

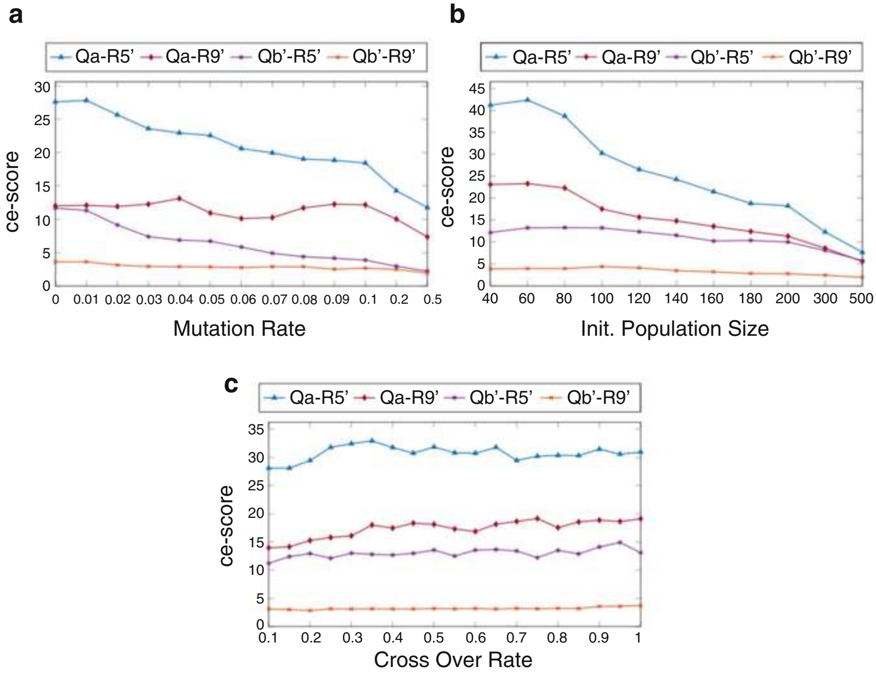


**Fig. 11.9** (**a**) CE-score over mutation rate, (**b**) CE-score over population size, (**c**) CE-score over crossover rate [274]

with 50 and 100 additional CESs, respectively. Hence, we have four test combinations on both simple and complex queries and repositories. The results of fine-tuning the mutation rate, population size and crossover rate are shown in Fig. 11.9a–c.

From the results in Fig. 11.9a, we can see that the optimal mutation rate is quite small for all tests, that is, from 0% to 0.4%. Results in Fig. 11.9b indicate that for smaller solutions spaces such as $Q_a$ over $R'_5$ and $R'_9$, the optimal initial population size is smaller, that is, with 60 individuals in the initial population. For more complicated queries and larger repositories, using a larger population size, for example, 100, is more cost-efficient. Results from Fig. 11.9c indicate that for $Q_a$ over $R'_5$, the optimal crossover rate is 35% because the global optimum is easier to achieve, and more crossover operations bring additional overhead. However, for more complicated queries and repositories, a higher crossover rate, for example, from 90% to 100%, is desired. It is worth noticing that in the results from Fig. 11.9b and c, the changes in the score for $Q'_b$ over $R'_9$ is not significant. This is because the GA spends much more time trying to initiate the population, making the cost-effectiveness score small and the differences moderate.

In the previous experiments, we use the selection policy such that every individual is chosen to reproduce once (except for the elite whose copy is also in the next-generation). This will ensure the population will get smaller as the evolution progresses and the GA will converge quickly. This is desirable in our case because the

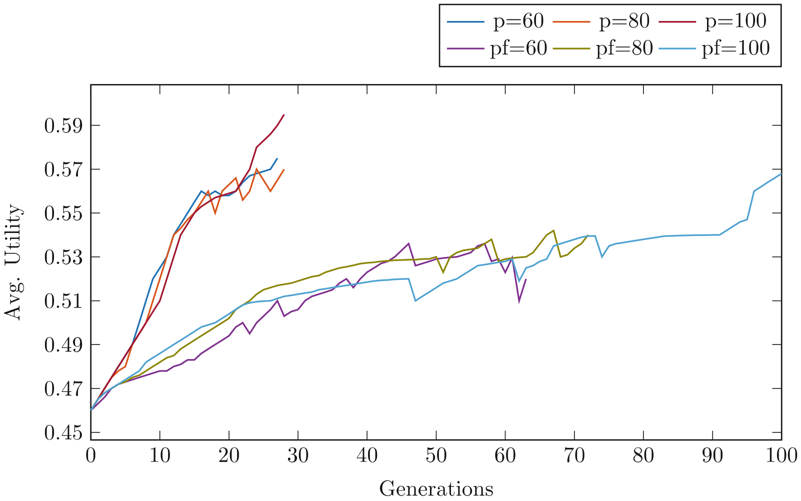


**Fig. 11.10** Average utility using flexible (“p = x”) and fixed (“pf = x”) population size [274]

algorithm is executed at run-time and is time sensitive. However, it is also possible to allow an individual to reproduce multiple times and keep a fixed population size during evolution.

To compare the differences of having a fixed or flexible population size, we show the average utility (of $Q'_b$ over $R'_9$) over the generations in Fig. 11.10. The results show that the number of generations for flexible population sizes is similar, while larger sizes achieve higher utilities. Also, the duration of generations in fixed population sizes is quite different: for a fixed population size of 60, the GA converges in about 60 generations; and for the size of 100, it lasts more than 100 generations. Larger sizes also produce better results in a fixed population, but it is much slower, and the utilities are lower than those obtained from flexible populations. In summary, we can confirm that using a flexible population size is better than a fixed population size for the GA described in this section.

### *11.5.2 Part 2: Validation of QoS Aggregation Rules*

In this part of the evaluation, we show how the QoS aggregation works in a simulated environment.

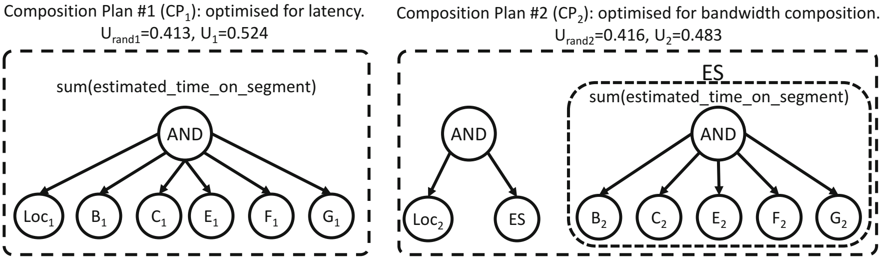

Composition Plan #1 ($CP_1$): optimised for latency.
$U_{rand1}$=0.413, $U_1$=0.524
sum(estimated_time_on_segment)
AND
$Loc_1$
$B_1$
$C_1$
$E_1$
$F_1$
$G_1$
Composition Plan #2 ($CP_2$): optimised for bandwidth composition.
$U_{rand2}$=0.416, $U_2$=0.483
AND
$Loc_2$
ES
ES
sum(estimated_time_on_segment)
AND
$B_2$
$C_2$
$E_2$
$F_2$
$G_2$

**Fig. 11.11** Composition plans for $Q_a$ under different weight vectors [274]

### 11.5.2.1 Datasets and Experiment Settings

To demonstrate the effect of QoS aggregation and optimisation, we generate two composition plans $CP_1$ and $CP_2$ with the GA for $Q_a$ over $R'_9$ using the same constraints specified in Sect. 11.5.1. $CP_1$ is optimised for latency, with the weight of latency set to 1.0 and other QoS weights set to 0.1; while $CP_2$ is optimised for network consumption, with the weight of network consumption set to 1.0 and others set to 0.1. The reason we generate one plan to reduce the latency and the other to reduce network consumption is that the resulting plans are quite different in structure, as shown in Fig. 11.11.

When the two composition plans are generated, we transform the composition plans into stream reasoning queries (e.g., C-SPARQL query). We evaluate the queries over the traffic data streams produced by ODAA sensors. According to the composition plan and the event service descriptions involved in the plans, we simulate the QoS of the event services on a local test machine, that is, we create artificial delays, wrong and lost messages according to the QoS specifications in event service descriptions, and set the sensor update frequency as the frequency annotated (so as to affect the messages consumed by the query engine). Security is annotated but not simulated, because the aggregation rule for security is trivial, that is, estimated to be the lowest security level. Notice that the simulated quality is the *Service Infrastructure* quality. We observe the results and the query performance over these simulated streams and compare it with the QoS estimation using the rules in Tables 11.1 and 11.2, to see the differences between the practical quality of the composed event streams and the theoretical quality as per our estimation.

### 11.5.2.2 Simulation Results

The results of the comparison between the theoretical and simulated quality of the event service composition are shown in Table 11.5. The first column is the quality dimensions of the two composition plans; the second column is the computed quality values based on the aggregation rules defined in Table 11.2. These rules consider the

**Table 11.5** Validation for QoS aggregation and estimation [274]

| | | Composition pattern | Event engine | End-to-end simulated | End-to-end deviations |
|---|---|---|---|---|---|
| Plan 1 ($CP_1$) | Latency | 40 ms | 604 ms | 673 ms | +4.50% |
| | Accuracy | 50.04% | 100% | 51.43% | +2.78% |
| | Completeness | 87.99% | 97.62% | 72.71% | −14.89% |
| | Network consumption | 4.05 msg/s | 4.05 msg/s | 3.84 msg/s | −5.19% |
| Plan 2 ($CP_2$) | Latency | 280 ms | 1852 ms | 2328 ms | +9.19% |
| | Accuracy | 53.10% | 100% | 51.09% | −3.79% |
| | Completeness | 87.82% | 73.18% | 46.31% | −17.96% |
| | Network consumption | 0.37 msg/s | 0.40 msg/s | 0.32 msg/s | −13.51% |

*Composition Pattern* of the query as well as the *Service Infrastructure* quality of the composed services. We denote this quality QoS. However, this is not the end-to-end QoS, because the quality of the event stream engine needs to be considered. To get the stream engine performance we deploy the queries with optimal *Service Infrastructure* quality, that is, no artificial delay, mistake, or message loss, and we record the quality of query executions in the third column. We denote this engine quality $QoS_{ee}$. The simulated end-to-end quality is recorded in the fourth column, denoted $QoS_s$. We calculate the theoretical end-to-end quality based on QoS and $QoS_{ee}$ using Table 11.1. Notice that the *Service Infrastructure* qualities of the queries themselves are not considered since we do not measure the results provided to external service consumers. Instead, the quality measurement stops at the point when query results are generated. We denote this theoretical end-to-end quality $QoS_t$ and calculate the deviation $d = (QoS_s/QoS_t) - 1$, which is recorded in the last column. From the results we can see that the GA is highly effective in optimising latency for $CP_1$ and network consumption for $CP_2$: the latency of the former is 1/7 of the latter and event messages consumed by the latter are less than 1/8 of the former.

We can also see that the deviations of latency and accuracy are moderate for both plans. However, the completeness estimation deviates about 15–18% from the actual completeness. For the network consumption in $CP_{1,}$ the estimation is quite accurate, that is, about 5% more than the actual consumption. However, the network consumption for $CP_2$ deviates from the estimated value by about 13.51%. The difference is caused by the unexpected drop in C-SPARQL query completeness when a CES with imperfect completeness is reused in $CP_2$, which suggests that an accurate completeness estimation of the service could help improve the estimation of the network consumption for event service compositions using the service.

## 11.6 Related Work

In general, Dataspaces follow a "best-effort" model for data management which can be seen as providing varying Quality of Service levels; this is evident in the area of search and querying with different mechanisms for indexing and federated queries [2, 121, 122]. In this section, we discuss the state-of-the-art in QoS-aware service composition as well as on-demand event/stream processing.

### *11.6.1 QoS-Aware Service Composition*

QoS models and aggregation rules for conventional web services have been discussed in [273, 277]. In this work, we extract some QoS properties from existing work and define a similar utility function. However, QoS aggregation in complex event services is different: the calculation of aggregated QoS depends on the correlations among member event services, while the impact of event engine performance also needs to be considered. Therefore, a set of new aggregation rules are developed in this work. GA-based service composition and optimisation have been explored previously in [273, 278, 279]. However, they only cater for *Input*, *Output*, *Precondition*, and *Effect* (IOPE) based service compositions. For composing complex event services, a pattern-based reuse mechanism is required [275].

### *11.6.2 On-Demand Event/Stream Processing*

Our work is not the first attempt that combines Service-Oriented Architectures (SOA) with Complex Event Processing to achieve on-demand event processing. Event-driven SOA has been discussed in [280, 281]. However, they only use CEP to trigger sub-sequential services. SARI [282] uses IOPE-based service matchmaking for event services, but it has limited matchmaking capability for logical AND and OR correlations.

A unified event query semantics is essential for a cross-platform and on-demand event processing [283]. EVA extends the semantic framework proposed by [284] and provides a transformation mechanism from EVA to target CEP query languages, but the transformation adaptation happens before run-time, and an on-demand event processing at run-time is not realised. RSQ-QL [283] is a recent unified RDF Stream Processing (RSF) query language that in theory supports event processing and existing RSP engines. However, it has yet to be implemented with a concrete engine.

Semantic Web Service inspired semantic streams [285] uses a prolog-based reasoning system to discover relevant data streams. It provides support for both functional and non-functional requirements, but the matchmaking still depends on the stream types. H2O [286] proposes a hybrid processing mechanism for long-term (coarse-grained) and real-time (fine-grained) queries, the former type of query can provide partial results for the latter. However, it limits the expressiveness of real-time queries. Dyknow [287] leverages C-SPARQL as its RSP engine and facilitates on-demand semantic data stream discovery using stream metadata annotations. However, Dyknow does not support complex event streams.

## 11.7 Summary and Future Work

In this chapter, we detail the design of a dataspace support service for the composition of a complex event service. The service uses a GA-based approach to find optimised event service compositions within a dataspace. A QoS aggregation schema is proposed to calculate the overall QoS for an event service composition based on correlations of member event services. A QoS utility function is defined based on the QoS model and serves as the objective function in the GA. Our algorithm is evaluated with both real and synthetic sensor data streams within an intelligent travel-planning system. Results show that we can achieve about 89% optimal results in seconds. We also provide experimental results on fine-tuning GA parameters to further improve the algorithm. Finally, we use experiments to validate our QoS aggregation schema, and the results indicate that our QoS aggregation and estimation do not deviate far from the actual QoS.

We are considering the following future directions. Firstly, we will explore rule-based event service composition as a more general approach for integrating various event services. The GA-based approach proposed in this work is an ad hoc solution in the sense that it relies on the event pattern semantics. Changing the event semantics might introduce significant revisions of the composition algorithm. Using a rule-based composition algorithm, on the other hand, can easily cope with various event semantics. Secondly, we will explore the distributed stream processing mechanisms for RDF streams and find efficient means to support dynamic reasoning in a distributed manner.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

# Chapter 12
# Dissemination of Internet of Things Streams in a Real-time Linked Dataspace

**Yongrui Qin, Quan Z. Sheng, and Edward Curry**

**Keywords** Linked data · Event processing · Stream dissemination · Event matching · Dataspaces · Internet of Things

## 12.1 Introduction

The Internet of Things (IoT) envisions smart objects and intelligent systems collecting and sharing data on a global scale to enable smart environments. One challenging data management issue is how to disseminate data to relevant consumers efficiently. This chapter leverages semantic technologies, such as linked data, which can facilitate machine-to-machine communications to build an efficient stream dissemination system for Semantic IoT. The system integrates linked data streams generated from various collectors and disseminates matched data to relevant consumers based on user queries registered in the system. We design two new data structures to suit the needs of high-performance linked data stream dissemination in the following two scenarios: (1) stream dissemination in point-to-point systems; and (2) stream dissemination in wireless broadcast systems. The evaluation of the approaches using real-world datasets shows that they can disseminate linked data streams more efficiently than existing techniques.

This chapter is structured as follows: Sect. 12.2 introduces the data management challenges of the IoT from the perspective of dataspaces. The design of the stream dissemination services is covered in Sect. 12.3. Section 12.4 discusses point-to-point linked data stream dissemination, and wireless broadcast is discussed in Sect. 12.5. Experimental evaluation of linked stream dissemination in the contexts of both point-to-point and wireless broadcast is discussed in Sect. 12.6. Section 12.7 discusses related work, with Sect. 12.8 summarising and highlighting future work.

© The Author(s) 2020

E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_12

## 12.2 Internet of Things: A Dataspace Perspective

The Internet of Things (IoT) aims to connect everyday objects, such as coats, shoes, watches, ovens, washing machines, bikes, cars, even humans, plants, animals, and changing environments, to the Internet to enable communications/interactions between these objects [288]. The goal of IoT is to enable computers to see, hear, and sense the real world. In the era of IoT, it is envisioned that smart objects and intelligent systems collect and share data on a global scale via the Internet.

In IoT, connecting all the things that people care about in the world becomes possible. All these Internet-connected things will produce vast scales of data in real time. Smart environments, leveraging IoT, are enabling data-driven intelligent systems that are transforming our everyday world, from the digitisation of traditional infrastructure (smart energy, water, and mobility) [289], the revolution of industrial sectors (smart autonomous cyber-physical systems, autonomous vehicles, and Industry 4.0), to changes in how our society operates (smart government and cities) [1]. Some promising IoT applications in future smart cities include resource management issues [290], effective urban street-parking management for reducing traffic congestion and fuel consumption [31, 291], efficient ways to distribute drinking water, assisting tracking and recovering stolen property [288], energy management [16], and so on.

On the Internet, the primary data producers and consumers are human beings. However, in the IoT, the main actors become *Things*, which means things are the main data producers and consumers. Therefore, in the context of the IoT, addressable and interconnected things, instead of humans, act as the primary data producers, as well as the primary data consumers. Intelligent systems will be able to learn and gain information and knowledge to solve real-world problems directly with the data fed from things. As a goal, intelligent systems within a smart environment enabled by IoT technologies will be able to sense and react to the real world for humans.

To make the potential of IoT a reality, we need to manage and process data efficiently and effectively, which will require new approaches to data management. Given the scale of data generated in IoT, topics such as distributed processing, real-time data stream analytics, and event processing are all critical. We may need to revisit these areas to improve upon existing technologies for applications in IoT scale [292, 293].

### *12.2.1 Real-time Linked Dataspaces*

To support the interconnection of intelligent systems in the data ecosystem that surrounds an IoT-based smart environment, there is a need to enable the sharing of data among systems. A data platform can provide a clear framework to support the sharing of data among a group of intelligent systems within a smart environment [1] (see Chap. 2). In this book, we advocate the use of the dataspace paradigm within the design of data platforms to enable data ecosystems for intelligent systems.

A dataspace is an emerging approach to data management that is distinct from current approaches. The dataspace approach recognises that in large-scale integration scenarios, involving thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources [2]. Within dataspaces, datasets *co-exist* but are not necessarily fully integrated or homogeneous in their schematics and semantics. Instead, data is integrated on an *as-needed* basis with the labour-intensive aspects of data integration postponed until they are required. Dataspaces reduce the initial effort required to set up data integration by relying on automatic matching and mapping generation techniques. This results in a loosely integrated set of data sources. When tighter semantic integration is required, it can be achieved in an incremental *pay-as-you-go* fashion by detailed mappings among the required data sources.

We have created the Real-time Linked Dataspace (RLD) (see Chap. 4) as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data/knowledge graphs and real-time stream and event processing capabilities to support a large-scale distributed heterogeneous collection of streams, events, and data sources [4].

Within the IoT, semantic technologies such as linked data, which aim to facilitate machine-to-machine communications, are playing an increasingly important role [294]. Linked data is part of a growing trend towards highly distributed systems, with thousands or potentially millions of independent sources providing structured data. These data sources can be managed within a dataspace for a smart environment. Due to a large amount of data produced within an IoT-based smart environment by different things, challenges exist with disseminating relevant data to multiple mobile data consumers in an efficient manner.

## 12.3 Stream Dissemination Service

The design of the stream dissemination service within the Real-time Linked Dataspace is based on our existing work in this area, including [122, 295, 296], which is brought together in this chapter and contextualised for use within the dataspace paradigm.

Figure 12.1 shows an overview of the RLD in a smart city scenario. We assume that data generated by all kinds of things will be represented in the form of linked data streams using the Resource Description Framework (RDF) and are managed within a dataspace. In Semantic IoT, the Semantic Sensor Network Ontology (SSN, http://www.w3.org/2005/Incubator/ssn/ssnx/ssn) can be used to model sensing data. Our service consists of two components: the *matching component* and the *index construction component*. Data consumers (humans and smart things in the city) can register their interests as user queries in the system. Then the index construction component will construct an index for all the user queries. The matching component will then evaluate incoming linked data streams against the constructed index to match triples with user queries efficiently. Finally, the system will disseminate matched data to relevant data consumers for further processing.

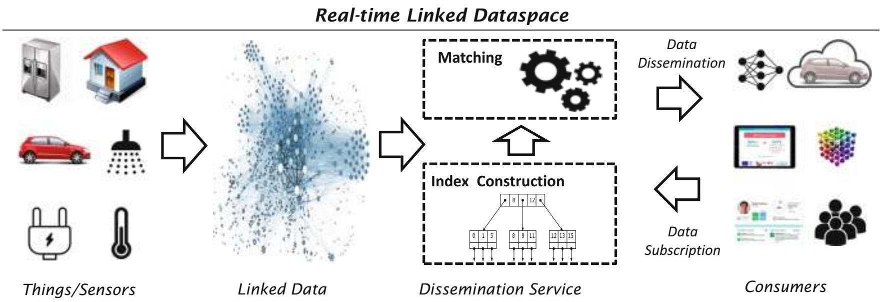


**Fig. 12.1** Overview of linked data stream dissemination service [295]

It is important to efficiently disseminate the most relevant data from IoT-based smart environments for the effective utilisation of the information streams. To this end, the Real-time Linked Dataspace supports data dissemination with a dedicated support service. The service uses two efficient data stream dissemination methods for semantic IoT enabled by linked data and semantic technologies.

- *Point-to-Point Stream Dissemination*: Supports the retrieval of relevant data from the deluge of the IoT data, which can then facilitate the extraction of useful information. The system first integrates the data generated from various data collectors. Then it transforms all the data to linked data streams (in RDF format, http://www.w3.org/RDF/). Meanwhile, data consumers can register their interest in the form of Basic Graph Patterns (BGPs, http://www.w3.org/TR/rdf-sparql-query/) in the system. Based on these BGPs, the system disseminates matched linked data to relevant users.
- *Stream Dissemination by Wireless Broadcasting*: Supports the massive number of mobile users in IoT-based smart environments. As the batteries of smart objects are often limited, an efficient way to reduce energy consumption and lower access latency is imperative. This can be achieved by designing effective and efficient air indexes for broadcasting linked data on air. We introduce a novel method by adopting 3D Hilbert curve mappings [297] for all the points converted from RDF triples. These mappings transform the 3D points into a sequence of one-dimensional points, which are suitable for efficient sequential access on air.

### 12.3.1 Pay-As-You-Go Service Levels

Dataspace support services follow a tiered approach to data management that reduces the initial cost and barriers to joining the dataspace. When tighter integration into the dataspace is required, it can be achieved incrementally by following the service tiers defined. The incremental nature of the support services is a core enabler of the pay-as-you-go paradigm in dataspaces. The functionality of the stream dissemination service follows the 5 Star pay-as-you-go model (detailed in

Chap. 4) of the RLD. It should be noted that currently, the stream dissemination service only supports events in linked data format. The stream dissemination service has the following tiered-levels of support:

*1 Star* **No Service:** No stream dissemination is supported.
*2 Stars* **No Service:** No stream dissemination is supported.
*3 Stars* **Point-to-Point:** Basic dissemination of streams between two points based on a simple matching of event sources using BGP. Services handle the transformation of streams to linked data at this level.
*4 Stars* **Wireless Broadcast:** Air indexes for broadcasting linked data within the smart environment.
*5 Stars* **Complex Patterns:** More expressive queries for complex event detection, in collaboration with the CEP service.

After receiving relevant data, users can further make use of the data to extract information for their purposes, such as environment monitoring, event detection, complex event processing, and so on. However, we will not discuss the data processing at the user side as this is the subject of Part IV of this book; in this chapter, we focus on how to match many BGPs against linked data streams efficiently.

## 12.4 Point-to-Point Linked Data Stream Dissemination

To disseminate high-quality information and provide high-performance matching services to data consumers (or subscribers) within a dataspace, we aim to design a system that will not return false-negative matched results. Therefore, we investigate pattern matching in this chapter. Pattern matching performs individual component matching between RDF triples and BGPs. It does not consider semantic relatedness between an RDF triple and a BGP (see Chap. 13 for semantic matching approaches). It may return false-positive matching results, but not false-negative ones. Recent works on pattern matching include linked data stream processing [298] and stream reasoning [299]. However, since these solutions are designed for optimisations of individual query evaluations, they are not suitable for processing many concurrent queries.

**User Queries** Basic Graph Patterns (BGPs) are adopted as user queries in our system. BGPs are sets of triple patterns. The possible triple patterns in a BGP are: (1) (#s, #p, #o), (2) (?s, #p, #o), (3) (#s, ?p, #o), (4) (#s, #p, ?o), (5) (?s, ?p, #o), (6) (?s, #p, ?o), (7) (#s, ?p, ?o), and (8) (?s, ?p, ?o). Here, ? denotes a variable, while # denotes a constant. Similar to data summaries [300], we apply hash functions (there are many different hash functions that are suitable for this purpose. For more details, please refer to [300]) to map these patterns into numerical values.

An example of pattern matching is that pattern (?s, :is, :Student) will match triple (:James, :is, :Student) but will not match (:James, :is, :PhDStudent). Other types of matching include match estimation and semantic matching, both of which may return false-negative results. Again, take pattern (?s, :is, :Student) as an example. In match estimation, the main task is to estimate which dataset matches pattern (?s, :is, : Student) the best by using some summarisation techniques among multiple datasets [300] to avoid querying all known datasets directly. In contrast, semantic matching will match semantically related triples compared to a specified pattern [151]. For example, pattern (?s, :is, :Student) may match (:James, :is, :PhDStudent) since the term:Student in the pattern is semantically related to :PhDStudent in the triple.

**Representations of Queries and Triples** In our linked data stream dissemination system, when the user queries (in the form of BGPs) are registered, all queries will be transformed into numerical values. The reason for this is that the comparisons between numbers are faster than strings. Note that we will have three numbers for the three components in a query as described above. Then a suitable index will be constructed for efficient evaluation between linked data streams and user queries. Before matching starts, RDF triples in the data streams will be mapped into numerical values. Then, these numerically represented triples will be matched with BGPs represented as numerical values in the constructed indexes.

### *12.4.1 TP-Automata for Pattern Matching*

Automata techniques have been adopted to process XML-based data streams [301]. They are based on languages with SQL-like syntaxes, and relational database execution models adapted to process streaming data. In our system, to support pattern matching, we apply automata to match each individual component of a triple with its counterparts of a BGP efficiently, which we call Triple Pattern automata (TP-automata).

Firstly, as mentioned, operating on numbers is more efficient than operating on strings. Note that when we map BGPs into numerical values, we treat variables in a BGP as a universal match indicator represented by "?". This indicator will be mapped into a fixed and unique numerical value but not the whole range of a specific coordinate axis. This unique numerical value will be treated differently as well later in the triple evaluation process.

Figure 12.2 depicts the construction process of TP-automata. Firstly, user queries will be transformed into triple pattern state machines, as shown in the middle of Fig. 12.2. As can be seen from the figure, each triple state machine contains an initial state, two internal states, one final state, and three transitions. In the figure, the first circle of a state machine represents the initial state, the next two circles represent the two internal states, and the doubled circle represents the final state. The three arrows associated with conditions are three transitions between different states. Similar to [301], these state machines can be combined into one machine by exploiting shared

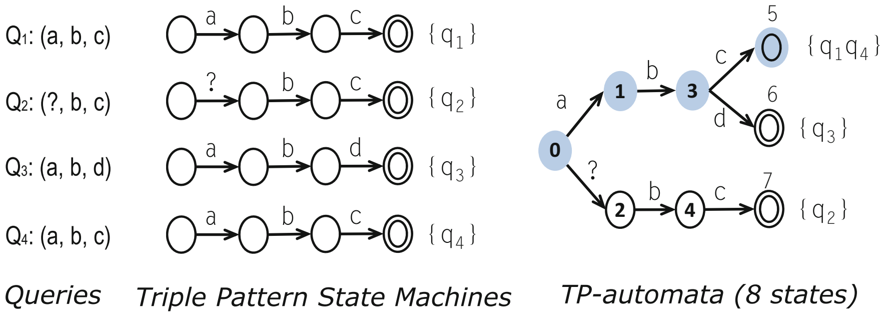


**Fig. 12.2** Query index structure: TP-automata [295]

common states with the same transitions. The combined machine, TP-automata, is shown on the right of Fig. 12.2. The shaded circles represent combined states.

To perform pattern matching over TP-automata, triples in the linked data stream will be first mapped into numerical values. For example, suppose a triple (s, p, o) is mapped into a 3D point (a, b, c). The system will match it against TP-automata in the following process. It first checks the initial state of TP-automata and looks for state transitions with the condition a or condition ?. Following the state transitions, state 1 and state 2 become the currently active states at the same time. It then looks for state transitions with condition b or ? from state 1 and state 2. Following the transitions, state 3 and state 4 become active states. Finally, following transitions with condition c or ? from state 3 and state 4, two final states, state 5 and state 7, are reached. By checking both final states, the system returns $\{q_1, q_2, q_4\}$ as the matching results. It should be noted that $q_3$: (a, b, d) will not match the input triple (a, b, c) as its object component's pattern is d, which does not match with c. The matching process stops if and only if all current active states are final states or states with no satisfied transition.

## 12.5 Linked Data Stream Dissemination via Wireless Broadcast

In a wireless data broadcast system, there is a base station that pre-processes data before it broadcasts the data on the wireless channel. If mobile clients have registered an interest in some data on the server, they can listen to the wireless channel and download the data. All mobile clients can share the wireless channel. In this way, the broadcast system could be able to serve an arbitrary number of mobile clients simultaneously.

For clients to efficiently locate data of interest, air indexing techniques are used to facilitate the searching of data on air. Air indexes usually are lightweight and concise summaries of the data to be broadcast. Based on air indexes, mobile clients within the communication range of the base station can evaluate their queries directly and then locate requested data on the wireless channel.

Similar to the existing work in data broadcast, we use access latency and tuning time as the primary performance metrics [302]. Access latency refers to the time elapsed between the moment when a query is formed, and the client starts listening to the server, to the moment when all requested data has been received. Tuning time refers to the period that a client must stay active to complete a query.

### 12.5.1 The Mapping Between Triples and 3D Points

Existing lightweight data summaries (e.g. [300, 303]) have proven to be effective to index linked data. However, they are not suitable in a wireless broadcast system. To develop a new index structure for broadcasting linked data, similar to data summaries, we choose to use hash functions to map RDF triples into numerical values. These numerical values can be regarded as coordinates in a 3D space. Precisely, given a hash function $F$, a triple (s, p, o) can be mapped into a 3D point ($F$ (s), $F$ (p), $F$ (o)). We call such a point mapped from a triple a data point to differentiate it from other points in the 3D space. Using this approach, a set of RDF triples can be mapped into a set of 3D data points.

BGPs are again used [303] to represent queries in our system. Like RDF triple mappings, a single BGP containing only one RDF triple pattern can be mapped into a point, a line, or a plane in a 3D space, or even the whole 3D space, depending on the number of variables in the triple pattern. The possible triple patterns in a BGP are: (1) (#s, #p, #o), (2) (?s, #p, #o), (3) (#s, ?p, #o), (4) (#s, #p, ?o), (5) (?s, ?p, #o), (6) (?s, #p, ?o), (7) (#s, ?p, ?o), and (8) (?s, ?p, ?o). Here, ? denotes a variable while # denotes a constant. Clearly, pattern 1 can be mapped into a 3D data point. Patterns 2 to 4 can be mapped into lines in the 3D space and patterns 5 to 7 can be mapped into planes. It should be noted that we do not consider pattern 8 in this approach, as it will be mapped into the whole 3D space and requires a traversal of all the data points in the whole 3D space, where air indexing is not required.

### 12.5.2 3D Hilbert Curve Index

A space-filling curve in D dimensions is a continuous, surjective mapping between one-dimensional space and D-dimensional space. A Hilbert curve is an example of a space-filling curve. It generally has good locality properties [297] and can efficiently support matching against BGPs with variables that can be mapped into lines or planes. Hence, the Hilbert curve is adopted as the foundation of our indexing method.

**Mapping 3D Points to One-Dimensional Points** To simplify our discussion, we use a 2D Hilbert curve to illustrate our ideas, which can then be generalised to 3D Hilbert curves. Figure 12.3 shows 2D Hilbert curves for order 1 and 2, that is, H1 and H2, respectively. Note that, a K order Hilbert curve, denoted as HK, passes all centre points of 2KD subdividing squares (or hypercubes) in a D-dimensional space. In Fig. 12.3a, each centre point of a subdividing square in 2D space is assigned a Hilbert value, which can be regarded as a one-dimensional point. Note that, the mapping between centre points and Hilbert values are bijective, which means for a given Hilbert curve, we can freely convert between centre points and Hilbert values in constant time.

From Fig. 12.3b, we can see that high-order Hilbert curves can be easily derived using transformation from low-order Hilbert curves like the one shown in Fig. 12.3b. To derive H2 from H1, in Fig. 12.3b, the 2D space is divided into 2D (D = 2 in this case) sub-regions, where each sub-region contains an H1. After rotating the lower two H1 curves, an H2 Hilbert curve is derived (see Fig. 12.3c). Since H1 has 22 subdividing squares, H2 has entirely subdividing squares.

Figure 12.4 presents an example of a 3D Hilbert curve of order 1. Higher-order 3D Hilbert curves can be derived using a similar process. To accommodate a larger 3D data space, that is, the hashing space for RDF triples, we need to utilise higher-order 3D Hilbert curves. We can quickly check that a K order 3D Hilbert curve can have up to 23 × K data points. In other words, when mapping to a K order 3D Hilbert curve, all RDF triples will be mapped into at most 23 × K data points (also centre points of hypercubes) in a 3D space.

**Indexing One-Dimensional Points on Air** After mapping 3D points into one-dimensional points using 3D Hilbert curves, we can utilise B+-trees to index one-dimensional points on a 3D Hilbert curve. An example is depicted in Fig. 12.5. Each one-dimensional point in the leaf nodes contains a pointer to a real triple that will be broadcast on the wireless channel. Such B+-trees can be serialised and broadcasted on the linear wireless channel as air indexes for the linked data on the air in the form of data packets. We adjust the fan-out of a B+-tree according to the packet capacity of the wireless channel so that a complete node of a B+-tree can fit in a packet. After downloading a part (e.g. a few packets) of an air index, mobile clients

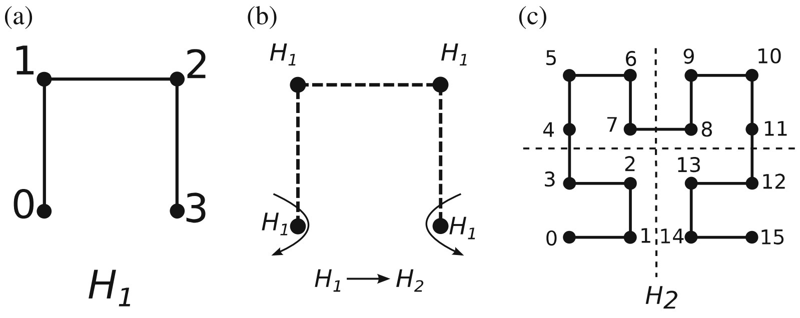


**Fig. 12.3** 2D Hilbert curves of order 1 and 2. (**a**) H1, (**b**) H1 to H2, (**c**) H2

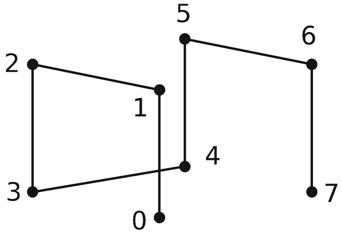


**Fig. 12.4** 3D Hilbert curve of order 1

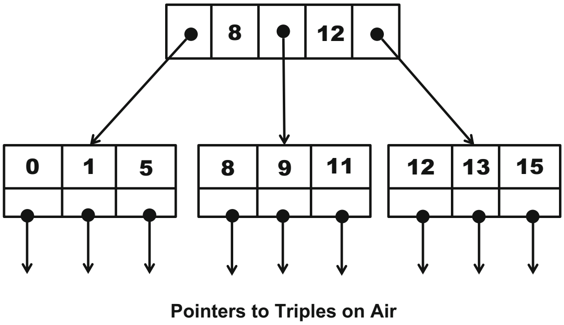


**Fig. 12.5** B+-tree for some points on a Hilbert curve

can then evaluate their queries (i.e. BGPs) against the partial index, determine which remaining index packets should be further retrieved, and finally compute the broadcast time of matched triples after all necessary index information has been acquired.

**Evaluating Queries Against an Air Index** In the query evaluation process, one challenging issue is how to match a one-dimensional point against BGPs. As mentioned previously, BGPs could be mapped into a point, a line, or a plane in a 3D space. In order to match BGPs with data points in the 3D space, we need to transform one-dimensional points in B+-tree based air indexes into 3D points. We then examine whether such 3D points fall into the subspace defined by a BGP. If yes, RDF triples pointed to by these points match the BGP. Otherwise, they do not match.

**Query Evaluation Example** Suppose a triple (a, b, c) can be hashed as (112, 31, 92) in a 3D space and its Hilbert value is 1137. Now suppose a mobile client issues a BGP (a, b, ?o). This BGP can be converted into a 3D line (112, 31, ?). After receiving Hilbert value 1137 from the air index, the mobile client firstly converts it back into a 3D point (112, 31, 92) and then it finds that this point falls on the 3D line defined by its BGP. Then the client knows the triple pointed to by Hilbert value 1137 is of its interest. As mentioned earlier, the conversion between a Hilbert value and a 3D point can be calculated in a constant time given a Hilbert curve of order K (here, K is a constant).

**Reducing Search Space** One issue needs to be addressed in the above query evaluation process: how to reduce the search space of Hilbert values indexed by B +-trees, thereby leading to fewer index packets required to download for query

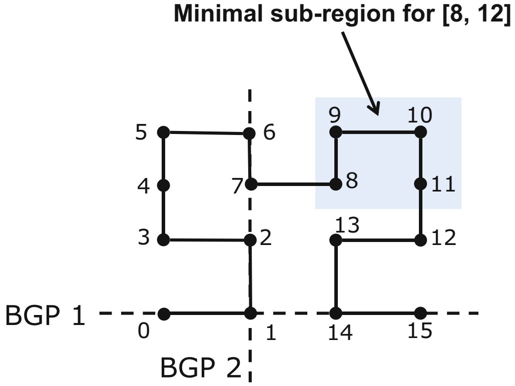


**Fig. 12.6** Minimal sub-region

evaluation. Given an air index like the one shown in Fig. 12.5, the root node has three child nodes. Based on the Hilbert values in the root node, we need to determine which child node would contain triples that may match a given BGP. We observe that each child node contains multiple Hilbert values, and the range of these values can be easily computed from the root node. For example, the value ranges of the three child nodes are [0, 8], [8, 12], and [12, HMAX] (here HMAX refers to the maximum Hilbert value of a Hilbert curve). For each value range, we have two bounding Hilbert values.

To reduce the search space, we compute the minimal sub-region defined by a lower-order Hilbert curve that covers the range defined by both bounding Hilbert values (see Fig. 12.6, where two dash lines represent two BGPs). If the minimal sub-region intersects with the sub-space (i.e. a line) defined by a BGP, the child node with the value range may contain triples that match that BGP. Otherwise, no triples in the child node will match that BGP. The example shown in Fig. 12.6 illustrates a minimal sub-region for the value range [8, 12]. We can see that two BGPs that are represented as two dash lines have no intersections with it. So, we can infer that no triples pointed to by the second child node (triples whose Hilbert values are 8, 9, and 11) in Fig. 12.5 will match the two BGPs shown as two dash lines in Fig. 12.6.

## 12.6 Experimental Evaluation

### *12.6.1 Evaluation of Point-to-Point Linked Stream Dissemination*

The dataset used in this experiment was generated in a smart office pilot, as discussed in Chap. 14 [100]. The energy readings were collected from August 4, 2014, to August 19, 2014. In total, there are around 6.2 million triples in the dataset. An event example is depicted in Table 12.1.

**Table 12.1** An event example [295]

| | | |
|---|---|---|
| @prefix do: <http://energy.deri.ie/ontology#> | | |
| @prefix dr: <http://../deri/deri rooms#> | | |
| :event1026fd7b0e5a | a | events:PowerConsumptionEvent . |
| :event1026fd7b0e5a | do:consumer | do:platform . |
| :event1026fd7b0e5a | do:consumerType | dr:Room01 . |
| :event1026fd7b0e5a | do:consumerLocation | dr:building01 . |
| :event1026fd7b0e5a | do:powerUsage | :usage9739ccddc76d . |
| :event1026fd7b0e5a | do:consumerDepartment | "facilities" . |
| :event1026fd7b0e5a | do:atTime | :timedb2c06100b33 . |
| :usage9739ccddc76d | a | dul:Amount . |
| :usage9739ccddc76d | do:hasDataValue | 171.87 . |
| :usage9739ccddc76d | do:isClassifiedBy | dr:watt . |
| :timedb2c06100b33 | a | do:Instant . |
| :timedb2c06100b33 | do:inDDateTime | "2014−08−12T18:17:18" . |

As an initial work, we used simple BGPs (i.e., single triple patterns) as queries in the experiment. We can simulate the join queries by letting data subscribers issue multiple simple BGPs. However, we leave extending our system to support complex BGPs or join queries as our future work. We randomly generated BGPs using the seven patterns mentioned in Sect. 12.4 based on our dataset. We did not consider the pattern (?s, ?p, ?o) in our experiment as it requires every triple in the linked data stream. In such a case, no query index is needed. We generated 10,000 queries to 100,000 queries.

We evaluated the performance of our approach in terms of Average Construction Time (in milliseconds) of the indexes and Average Throughput (number of triples per second). We implemented hash-based TP-automata (i.e. we map triples and queries into numerical values and denote such method as *HashMat* in the following figures) and string-based TP-automata (i.e. we use triples and queries as is and denote this method as *StringMat* in the following figures). We compared HashMat and StringMat with state-of-the-art pattern matching technique, CQELS [298] (https://code.google.com/p/cqels/), which is also designed for linked data streams. We examined the matching quality of the hash-based TP-automata as well. All methods were implemented on Java Platform Standard Edition 7 running on Linux (Ubuntu 12.10, 64-bit Operating System), with quad-core CPU@2.20GHz and 4 GB memory. We ran each experiment 10 times and reported their average experimental results.

The performance of pattern matching on TP-automata is presented in Fig. 12.7. Average construction time is compared in Fig. 12.7a. The construction times for both hash-based TP-automata and string-based TP-automata are similar to each other in most settings. For more significant numbers of queries, such as 75k and 100k queries, the construction of string-based indexes takes a slightly longer time. Usually, the construction can be completed within a few hundred milliseconds. However, the construction time of CQELS takes much longer, which usually requires around ten thousand milliseconds.

Throughput performance of pattern matching is depicted in Fig. 12.7b. It shows some substantial differences between CQELS and TP-automata based approaches (HashMat and StringMat). Generally, HashMat and StringMat can achieve throughput at the speed of nearly a million triples per second and are about four orders of magnitude faster than CQELS. The main reason for this is that CQELS is a much more comprehensive system focusing on optimising evaluation of queries with complex operators and semantics but not on the evaluation of a large set of concurrent and straightforward queries over linked data streams. In this regard, our approach can also be adapted to complement CQELS for dealing with our linked data stream dissemination scenario. Regarding HashMat and StringMat, in most cases, HashMat achieves about twice the throughput speed compared with StringMat.

Finally, we investigated the matching quality of hash-based TP-automata (HashMat) via Precision, Recall, and $F_1$ Score. This is because collisions are difficult to avoid in any hash-based approaches, and false-positives exist in hash-based TP-automata, which affects matching quality. Specifically, we investigated Precision and $F_1$ Score when the Recall is 100% since we observe that the matching quality of HashMat is already excellent in such cases. As can be seen in Table 12.2, the Precision and $F_1$ Score are 100% when the number of queries is 10k or 25k. For more significant numbers of queries (e.g. 50k, 75k, and 100k), both Precision and $F_1$ Score are still higher than 99.99950%. This demonstrates that HashMat provides a very high matching quality.

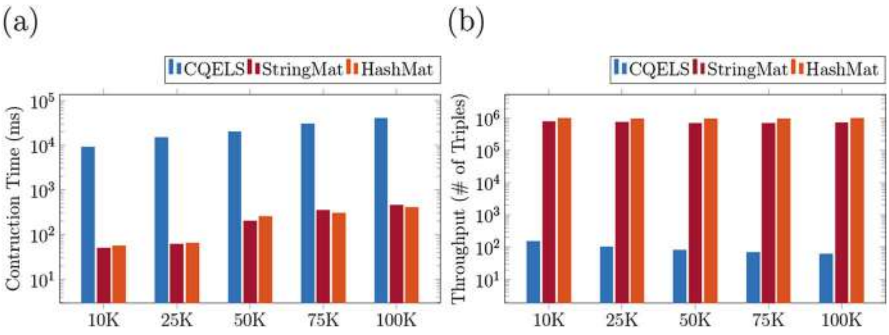


**Fig. 12.7** Performance on pattern matching. (**a**) Average construction time. (**b**) Average throughput [295]

**Table 12.2** Matching quality of HashMat when the recall is 100%

| Queries | Recall (%) | Precision (%) | $F_1$ Score (%) |
|---|---|---|---|
| 10k | 100 | 100 | 100 |
| 25k | 100 | 100 | 100 |
| 50k | 100 | 99.99975 | 99.99987 |
| 75k | 100 | 99.99982 | 99.99991 |
| 100k | 100 | 99.99960 | 99.99980 |

### 12.6.2 Evaluation on Linked Stream Dissemination via Wireless Broadcast

The dataset used in this experiment is a subset of the current version of the English DBpedia.[1] It contains resources of type dbpedia-owl:Event. Each event is a triple of the form <eventURI, rdf:type, dbpedia-owl:Event>. An example of an event URI is http://dbpedia.org/resource/Battle_of_Brentford_(1642). There are approximately 400,100 triples in the dataset.

As an initial work, we used simple BGPs as queries in the experiment, and we leave extending our system to support complex BGPs or join queries as our future work. We randomly generated BGPs using the seven patterns mentioned in Sect. 12.5.1 based on our dataset. We generated 100,000 queries and reported the average experimental results.

We compared our B+-tree HC (Hilbert Curve) indexes with traditional R-tree indexes, which can be used to index 3D points directly. In the experiment, we varied the packet capacity of the wireless broadcast channel from 128 bytes to 2048 bytes. For each packet capacity setting, we assigned appropriate fan out and leaf order parameters for R-trees and B+-trees to ensure that each packet was able to accommodate a complete node of a tree.

Figure 12.8a shows the comparisons of the access latency and Fig. 12.8b shows the comparisons of the tuning time. From the figure, we can identify that the HC-based index outperforms the R-tree based index. The reason for this is twofold: firstly, by using our novel search algorithm, the search space of the HC-based index is smaller than the R-tree-based index; secondly, the size of each index entry for the HC-based index is smaller than the R-tree-based index. This result confirms the effectiveness and efficiency of our search algorithm and HC-based indexing technique.

The percentage of index tuning time is presented in Fig. 12.9a. Here, we define the percentage of index tuning time as the ratio between the index tuning time (caused by downloading necessary parts of the index on air) and the total tuning time (the sum of index tuning time and content tuning time). This metric is a good indicator of the effectiveness and efficiency of an index. The lower the percentage we get, the better is the effectiveness we can achieve. Figure 12.9a shows that the HC-based index has a much lower percentage of index tuning time when compared with the R-tree-based index. To be specific, the percentage of index tuning time of the HC-based index is below 20% under different packet capacities while that of the R-tree-based index is above 60%.

The index sizes are compared in Fig. 12.9b. We can see that the number of packets required to accommodate the whole index for HC-based index is only about half of that for R-tree-based index. The main reason is that the R-tree index must store 3D points in its nodes while the HC-based index only stores one-dimensional points.

[1] http://downloads.dbpedia.org/3.8/en/

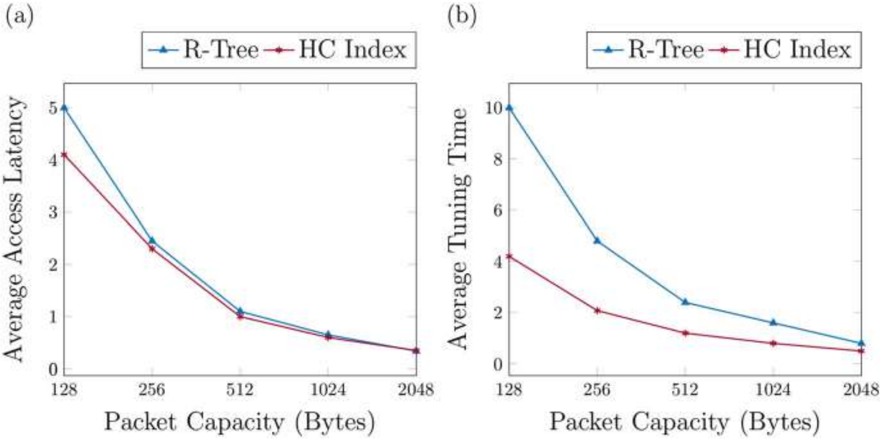


**Fig. 12.8** Performance evaluation for (**a**) access latency and (**b**) tuning time

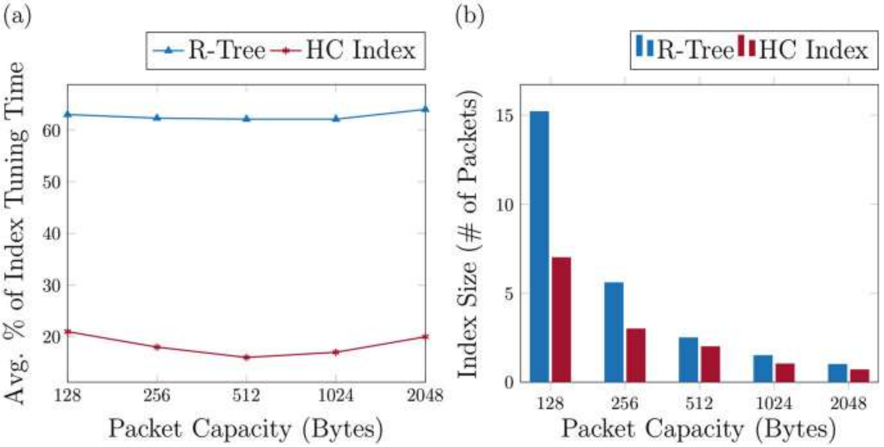


**Fig. 12.9** Performance evaluation for (**a**) index tuning time and (**b**) index size

## 12.7 Related Work

### *12.7.1 Matching*

Recent work on data summaries on linked data [300] transform RDF triples into a numerical space. Then data summaries are built upon numerical data instead of strings, as summarising numbers is more efficient than summarising strings. To transform triples into numbers, hash functions are applied to the individual components (s, p, o) of triples. Thus, a derived triple of numbers can be considered as a 3D point. In this way, a set of RDF triples can be mapped into a set of points in a 3D space. To facilitate query processing over data summaries, a spatial index named QTree [300], which is evolved from standard R-trees [304], is adopted as the basic index. Data summaries are designed for indexing various linked data sources and used for identifying relevant sources for a given query.

However, data summaries are not suitable for our linked data stream dissemination systems. Firstly, techniques on data summaries, such as QTree, do not consider variables in the BGPs but only RDF triples with concrete strings. Further, since data summaries are concise and imprecise representations of data sources [300], they provide match estimation. Hence, query evaluation on them would return false-negative results, which is not allowed in our system.

Semantic matching has also been studied, which aims to match semantically related RDF triples against BGPs. It may provide false-positive match results but not a false-negative. Both approximate event matching [151] and thematic event processing [272] apply semantic matching. Similarly, all these techniques will return false-negative matching results, which are not allowed in our approach.

Moreover, existing work on pattern matching, such as stream reasoning [299] and linked data stream processing [298], does not support large-scale query evaluation but focuses on the evaluation of a single query or a small number of parallel queries over the streaming linked data. Therefore, the issue of supporting pattern matching over a large number of BGPs against linked data streams remains open.

### *12.7.2 Wireless Broadcast*

In order to support efficient query processing among a large number of data sources, relevant sources that can better answer the queries should be identified first, and then the queries are evaluated on them. Lightweight data summaries on linked data [300] have been investigated to determine relevant sources to use during query evaluation. However, these techniques are mainly designed for random access in memory or on disk, and thus they cannot be applied in a wireless broadcast system directly, where only linear access is allowed.

The work of CkNN (Continuous k Nearest Neighbor) query processing on air [305] has introduced indexes and search algorithms for continuous kNN queries in 2D spaces. It uses a similar technique discussed in this chapter to index moving objects modelled in 2D spaces. Their work differs from this work as their main goal is to support continuous kNN queries in 2D spaces, while the main goal of this chapter is to support queries on linked data broadcast in a wireless channel.

Our work differs from the CkNN query processing in the following aspects: (1) CkNN query processing is for CkNN queries while our work aims at processing Basic Graph Patterns (BGPs) on linked data; (2) the search algorithms are different as query evaluation for CkNN queries, and BGPs is different; (3) the indexing space is different as CkNN query processing only considers 2D space but our work considers 3D space; (4) finally, our work can be extended to support more complex queries on linked data broadcast in a wireless channel while CkNN query processing aims to handle variants of NN queries.

## 12.8 Summary and Future Work

In this chapter, we have leveraged semantic technologies, such as linked data, to build an efficient stream dissemination system for semantic IoT within smart environments. We present the design of the stream dissemination support service for a dataspace based on two new data structures to suit the needs of high-performance linked data stream dissemination in (1) point-to-point systems; and (2) wireless broadcast systems.

In order to efficiently match a large number of BGPs against linked data streams, we have proposed TP-automata, an automata-based method designed for efficient pattern matching. In our evaluation, we show that TP-automata can disseminate linked data within the dataspace at the speed of nearly one million triples per second with 100,000 registered user queries and is several orders of magnitude faster in terms of both index construction time and throughput compared with the state-of-the-art technique. Further, using hash-based TP-automata, the throughput is doubled compared with string-based TP-automata with high matching quality.

We have proposed an effective and efficient air indexing method, HC-Index, for broadcasting linked data on air, which can be used in data sharing among a large number of mobile, smart objects, and intelligent systems in an IoT-based smart environment. Our method is based on 3D Hilbert curve mappings. Firstly, we map RDF triples into points in a 3D space and then adopt 3D Hilbert curve mappings to convert all the 3D points into one-dimensional points. We build B+-trees upon these one-dimensional points and serialise these trees to accommodate them on the linear wireless channels. An efficient search algorithm has been devised to facilitate query processing over the linked data on air. We have conducted experiments and compared our method with the traditional R-tree-based spatial indexing method. Our method has shown better performance over the R-tree-based method in various aspects, including access latency, tuning time, and index size.

Future work includes supporting more complex user queries, such as join queries; and supporting semantic matching in a hashing space with the use of Locality-Sensitive Hashing (LSH) techniques [306] that help to map semantically related data together. Both directions will enable the linked data stream dissemination system to provide better semantic richness and to support data consumption needs more accurately, which is a critical issue in the IoT. We will also consider the challenges of matching and dissemination of multimedia events [30] within smart environments.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

# Chapter 13
# Approximate Semantic Event Processing in Real-time Linked Dataspaces

Souleiman Hasan and Edward Curry

**Keywords** Event processing · Semantic matching · Best-effort · Internet of things · Dataspaces

## 13.1 Introduction

Within dataspaces, data sources are not necessarily fully integrated or homogeneous in their schematics and semantics. For dataspaces to support a real-time response to situations of interest when a set of events take place, for example from sensor readings, there is a need for a principled approach to tackling data heterogeneity within real-time data processing. In this chapter, we detail techniques for developing dataspace support services for dealing with the semantic heterogeneity of real-time data.

In the following sections, we build upon the discussion so far in this book and focus on best-effort semantic matching for real-time data processing in dataspaces (Sect. 13.2). In Sect. 13.3, we discuss an approximate semantic matching service for real-time data processing in dataspaces, represented by the approximate semantic and thematic event processing models. The elements of the approach are detailed in Sect. 13.4. Then, in Sect. 13.5, we discuss an instantiation of these event processing models, with their evaluation in Sect. 13.6. We finish with an analysis of related work in Sect. 13.7, and then conclude the chapter in Sect. 13.8.

## 13.2 Approximate Event Matching in Real-time Linked Dataspaces

Real-time data sources are increasingly forming a significant portion of the data generated in dataspaces. This in part is due to increased adoption of the Internet of Things (IoT) and the use of sensors for improved data collection and monitoring of

© The Author(s) 2020
E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_13

daily activities in smart buildings, smart energy, smart cities, and others. In this section, we explore the concepts of dataspaces and event processing to understand data management challenges in dataspaces with real-time data sources. We then introduce the approximate semantic event matching services to address the challenge of semantic matching of heterogeneous events. The design of the service is based on our existing work in approximate semantic event matching, including [151, 272, 307, 308], which is brought together in this chapter and contextualised for use within the dataspace paradigm.

### *13.2.1 Real-time Linked Dataspaces*

Driven by the adoption of the IoT, smart environments are enabling data-driven intelligent systems that are transforming our everyday world, from the digitisation of traditional infrastructure (smart energy, water, and mobility), the revolution of industrial sectors (smart autonomous cyber-physical systems, autonomous vehicles, and Industry 4.0), to changes in how our society operates (smart government and cities). To support the interconnection of intelligent systems in the data ecosystem that surrounds a smart environment, there is a need to enable the sharing of data among systems. A data platform can provide a clear framework to support the sharing of data among a group of intelligent systems within a smart environment [1] (see Chap. 2). In this book, we advocate the use of the dataspace paradigm within the design of data platforms to enable data ecosystems for intelligent systems.

A dataspace is an emerging approach to data management which recognises that in large-scale integration scenarios, involving thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources [2]. The dataspace paradigm pushes the boundaries of traditional data management approaches in two main dimensions [2]: *Administrative Proximity*, which describes how data sources within a space of interest are close or far in terms of control; and *Semantic Integration*, which refers to the degree to which the data schemas within the data management system are matched up. These dimensions form part of the three boundaries (knowledge, value, and ecosystem from the Knowledge Value Ecosystem (KVE) Framework) that need to be crossed in order for knowledge exchange to occur among systems within a data ecosystem (see Chap. 2 for further discussion on this topic).

Within dataspaces, data sources are not necessarily fully integrated or homogeneous in their schematics and semantics. Instead, data is integrated on an *as-needed* basis with the labour-intensive aspects of data integration postponed until they are required. Dataspaces reduce the initial effort required to set up data integration by relying on automatic matching and mapping generation techniques. This results in a loosely integrated set of data sources. When tighter semantic integration is required, it can be achieved in an incremental *pay-as-you-go* fashion by detailed mappings among the required data sources.

We have created the Real-time Linked Dataspace (RLD) (see Chap. 4) as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data, knowledge graphs, and real-time stream and event processing capabilities to support a large-scale distributed heterogeneous collection of streams, events, and data sources [4]. In order to understand the requirements of real-time data processing, we will explore the event processing paradigm [139].

### *13.2.2 Event Processing*

In event processing, data items that are shared within the dataspace in real time are called *events*. Data sources which produce events are called *event producers*. Users and software which are interested in an event, or set of events, are called *event consumers*. For example, in a smart building, there can be IoT-based data sources which produce information continuously, such as energy consumption sensors and motion detection sensors within an office. Data items produced by such sensors are the events. The sensor is the event producer. A building manager may be interested in situations where a light in an office is left on while the office is unoccupied. In this example, the building manager and the software representation of their interest (the event query) would be the event consumer.

Thus, an essential part of the event processing paradigm is the matching mechanism between the events and the interests of event consumers. This is similar to the concept of query processing in relational database management systems, where events replace the concept of a data tuple, and subscriptions or rules replace the concept of queries. In a specific family of event processing systems, called stream processing, queries take the name of continuous queries as they are evaluated continuously against data.

In terms of crossing system boundaries for data sharing, the decoupled nature of event-based systems reduces their administrative proximity. However, in terms of semantic integration, event-based systems currently require tight semantic integration [151]. Acknowledging that the challenges to dataspaces such as loose administrative proximity and loose semantic integration can face real-time data sources, the question becomes: how can we support the loose administrative proximity and semantic coupling within real-time data processing? To answer this question, we look no further than the literature of the event processing paradigm itself. A core principle in event processing is decoupling, which refers to the lack of explicit agreements in order to increase scalability as defined by Eugster et al. [142]. Three

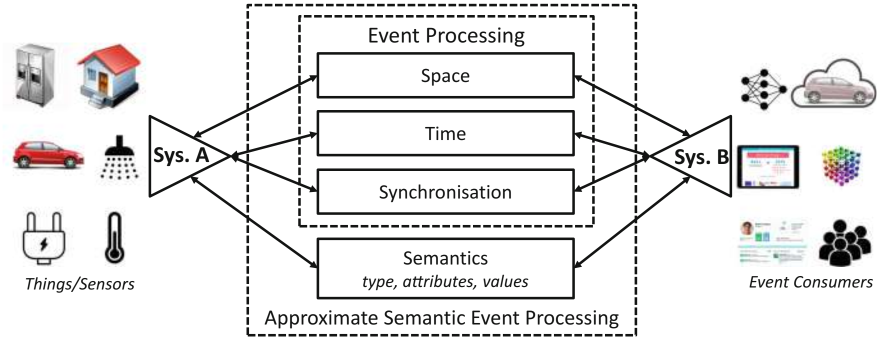


**Fig. 13.1** The decoupling dimensions of event processing (Adapted from [151])

main dimensions have been recognised in the event processing literature concerning decoupling, as illustrated in Fig. 13.1:

- *Space Decoupling*: Which means that event producers and consumers do not hold addresses, such as IPs, of each other.
- *Time Decoupling*: Which means that event producers and consumers do not have to be active at the same time.
- *Synchronisation Decoupling*: Which means that event producers and consumers do not block each other when exchanging events.

We build on the concept of decoupling to meet the principles of Real-time Linked Dataspaces as detailed in Chap. 4, that is, an event processing paradigm that supports many formats of data, does not depend on schema agreement, and supports a best-effort approximate and pay-as-you-go approach. We identify this as a new dimension for event processing systems which we call *loose semantic coupling*.

## 13.3 The Approximate Semantic Matching Service

Loose coupling of event processing systems on the semantic dimension reflects a low cost to define and maintain rules concerning the use of terms, and a low cost for building and agreeing on the event semantic model. This requirement forms the foundation of our semantic matching models and their enabling elements, as discussed in the remainder of this chapter.

### *13.3.1 Pay-As-You-Go Service Levels*

Dataspace support services follow a tiered approach to data management that reduces the initial cost and barriers to joining the dataspace. When tighter integration into the dataspace is required, it can be achieved incrementally by following the service tiers defined. The incremental nature of the support services is a core enabler of the pay-as-you-go paradigm in dataspaces. The semantic matching models have been used within an approximate semantic event processing support service within the RLD. The functionality of the service follows the 5 Star pay-as-you-go model (detailed in Chap. 4) of the RLD. The approximation semantic matching service has the following tiered-levels of support:

*1 Star* **No Service:** No semantic matching is supported.

*2 Stars* **Semantic Matching:** Approximate semantic matching at the attribute-value of events and subscriptions.

*3 Stars* **Thematic Matching:** Thematic matching of events with the use of theme tags to more accurately describe events in a low-cost manner.

*4 Stars* **Entity-Centric:** Event matching is performed over entity-centric event graphs (e.g. RDF).

*5 Stars* **Context-Aware:** Context-aware semantic matching of events with the use of external knowledge from the dataspace.

### *13.3.2 Semantic Matching Models*

The main semantic matching models and elements of the approach are presented, with respect to the event flow model presented by Cugola and Margara in [139]. The core components of an event engine in their model are the event *Receiver*, the *Decider*, the *Producer*, and the event *Forwarder*. Event *Sources*, *Consumers*, and *Users* interact with the engine through protocols and condition/action *Rules*. Figure 13.2 presents an elaboration of Cugola and Margara's model with additional models and elements for approximate semantic event processing. Two models form the basis for the approach: (I) the approximate event matching model, and (II) the thematic event matching model, which are outlined in the following sections.

### *13.3.3 Model I: The Approximate Event Matching Model*

This model extends the current event processing paradigm through the following:

- Event processing rules are equipped with the *tilde* $\sim$ semantic approximation operator so users can express their delegation to the event engine to match similar, or related, event terms, to the term used in a subscription. The background semantic model for approximation is a statistical model built from co-occurrences of terms in a large corpus of plain text documents. For instance, the following subscription tells the event engine to match it to events generated with device equal to *'laptop'* or similar terms and to match the value "room 112" with the term *'office'* or related terms such as *'room'* or *'zone'*.

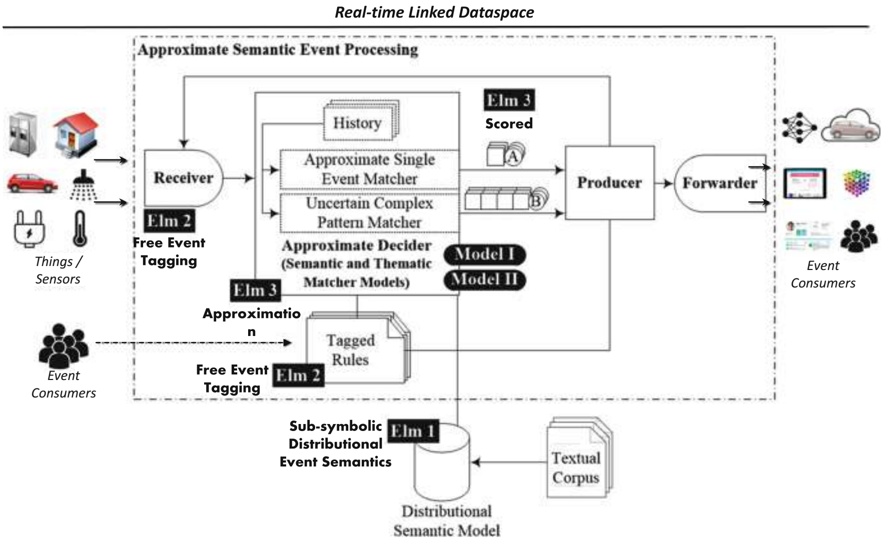

Real-time Linked Dataspace
Approximate Semantic Event Processing
Things / Sensors
Receiver
Elm 2
Free Event Tagging
History
Approximate Single Event Matcher
Uncertain Complex Pattern Matcher
Approximate Decider (Semantic and Thematic Matcher Models)
Model I
Model II
Elm 3
Scored
A
B
Producer
Forwarder
Event Consumers
Elm 3
Approximation
Tagged Rules
Free Event Tagging
Elm 2
Event Consumers
Sub-symbolic Distributional Event Semantics
Elm 1
Textual Corpus
Distributional Semantic Model

**Fig. 13.2** Models and elements in the approximate semantic event processing model

$$\begin{aligned}&\{\textbf{type} = \text{increased energy usage event},\\ &\textbf{device} = \text{laptop} \sim,\\ &\textbf{office} \sim= \text{room 112}\}\end{aligned}$$

- The single event matcher is equipped with matching and mapping algorithms to detect events semantically relevant to approximate subscriptions. For instance, let an event of *increased energy consumption* be represented as follows:

$$\begin{aligned}&\{\textbf{type} : \text{increased energy consumption event},\\ &\textbf{measurement unit} : \text{kilowatt-hour},\\ &\textbf{device} : \text{computer},\\ &\textbf{office} : \text{room 112}\}\end{aligned}$$

The most probable mapping, or the top−1 mapping, of this event to the previous subscription is generated as a probable scored result. It can be described as follows:

$$\begin{aligned}\sigma^* = \{&(\textbf{type} = \textbf{increased energy consumption event}\\ &\leftrightarrow \text{type} : \text{increased energy usage event}),\\ &(\textbf{device} \sim= \textbf{laptop} \sim\leftrightarrow \text{device} : \text{computer}),\\ &(\textbf{office} = \textbf{room}\ \ \textbf{112} \leftrightarrow \text{office} : \text{room}\,112)\}\end{aligned}$$

- The complex pattern matcher can then perform probabilistic reasoning to deduce the probabilities of occurrences of the derived events in the action parts of the complex rules.

### 13.3.4 Model II: The Thematic Event Matching Model

This model suggests associating free tags that describe the themes of types, attributes, and values in events and subscriptions, in order to clarify their meanings. For instance, the previous *increased energy consumption* event is associated with tags as follows:

$$\{\text{appliances, building}\}$$

These tags help disambiguate the meaning of terms in the event such as *'energy'* and *'office'* and move them closer to the energy management domain in smart buildings. Thematic events can more easily cross semantic boundaries as (1) they free users from needing a prior semantic top-down agreement, and (2) they carry approximations of events' meanings composed of payloads and theme tags which, when combined, carry less semantic ambiguities. An approximate matcher exploits the associated thematic tags to improve the quality of its uncertain matching of events and subscriptions.

## 13.4 Elements for Approximate Semantic Matching of Events

The approach can be conceptually decomposed into three main elements, as outlined in the following sections and illustrated in Fig. 13.2.

### 13.4.1 Elm 1: Sub-symbolic Distributional Event Semantics

This element stems from the need for loosening the semantic coupling between event producers and consumers. If semantic coupling can be quantified by the number of mappings between symbols, that is, terms, and meanings, then a semantic model that condenses these mappings can be particularly useful. Ontological models require granular agreements on the symbol-meaning mappings, which is proportional to the number of symbols. However, distributional vector space semantics leverage the statistics of terms co-occurrence in a large corpus to establish semantics [309]. For instance, the terms *'power'* and *'electricity'* would frequently co-appear in an energy management domain corpus. Thus, they can be assumed to be related, and this can be leveraged within energy event matching. Using such a model leaves event producers and consumers to loosely agree on the corpus as a representative of their common knowledge and decrease the need for granular agreements on every individual term of the domain.

### 13.4.2 Elm 2: Free Event Tagging

This element stems from the need to enable event processing within a loosely coupled model to effectively and efficiently allow users to adapt the conveyed events' meanings in different domains and situations. Free tagging of events and subscriptions do not introduce any coupling components between participants, in contrast to the case of top-down fixed taxonomies. This element builds on the success of free tagging, known as folksonomies, within social media research [310]. For instance, the term *'energy'* when used in an event tagged by the tags {*'building'*, *'appliance'*} helps the matcher distinguish the meaning of *'energy'* and associate it with the domain of power management, rather than associating it with the domain of sport or diet, for example. When used to process IoT events, free tagging can create "Thingsonomies" as a way to support the discovery and use of events from Things [308].

### 13.4.3 Elm 3: Approximation

This element stems from the realisation that loosening the coupling between event producers and consumers at the semantic and pragmatic levels introduces uncertainties to the engine. Uncertainty results from not exactly knowing which event's tuples shall be mapped to which subscription's tuples. For instance, with the loose agreements on terms' semantics, there are various possible mappings between an event and a subscription such as:

$$\begin{aligned}\sigma_1 = \{&(\textbf{device} = \textbf{laptop} \leftrightarrow \text{device : computer}),\\ &(\textbf{room} = \textbf{room\ \ 112} \leftrightarrow \text{office : room 112})\}\end{aligned}$$

$$\begin{aligned}\sigma_2 = \{&(\textbf{device} = \textbf{laptop} \leftrightarrow \text{office : room 112}),\\ &(\textbf{room} = \textbf{room 112} \leftrightarrow \text{device : computer})\}\end{aligned}$$

Each mapping has a different probability which reflects the uncertainty of the matching. Approximation at the core of the event processing engine can tackle uncertainties and complement the elements mentioned earlier.

### 13.4.4 Elements Within the Event Flow Functional Model

The main elements of the approach can be unified and placed into the event processing functional model illustrated in Fig. 13.2 as follows:

- *Elm1—Sub-symbolic Distributional Event Semantics*: The actual distributional semantic model could be built outside of the event processing engine by indexing a textual corpus. The resulting model forms the basis to compare any two strings in events and subscriptions, as they get decoded into their vector representations. Vectors form the basis for distance and similarity measures.
- *Elm2—Free Event Tagging*: Events are flowing from event sources, and subscriptions get tagged by users before they are considered for matching. Users use free tags to enhance events and subscriptions and improve their interpretation by the matcher.
- *Elm3—Approximation*: Events are matched in the decider against subscriptions. The decider is now approximate, and the result of matching is represented by scored events (Elm3-*Scored*) which signify their relevance to each subscription. The decider makes use of the semantic model and the tags when matching the events.

## 13.5 Instantiation

The instantiation of the models requires a concrete model for the events, the subscription language, and the matching model, as discussed in the following sections.

### *13.5.1 Events*

The most elaborate event model is the instantiation of the thematic event model, which is based on an attribute-value model. Each event is a pair of sets: a set of theme tags and a set of tuples. Each theme tag is a single word or a multi-word term. Each tuple consists of an attribute-value pair. No two distinct tuples can have the same attribute. An example energy consumption event is represented as follows:

$$
\begin{aligned}
&(\{\textbf{energy}, \textbf{appliances}, \textbf{building}\},\\
&\{\textbf{type} : \text{increased energy consumption event},\\
&\textbf{measurement unit} : \text{kilowatt-hour},\\
&\textbf{device} : \text{computer},\\
&\textbf{office} : \text{room } 112\})
\end{aligned}
$$

The formal definition of the event model is as follows: let $E$ be the set of all events, let $TH$ be the set of all possible theme tags, and let $A$ and $V$ be the sets of possible attributes and values respectively. Let $AV$ be the set of possible attribute-value pairs, that is, tuples, such that $AV = \{(a, v): a \in A \wedge v \in V\}$. An event $e \in E$ is a pair $(th, av)$ such that $th \subseteq TH$ and $av \subseteq AV$ are the set of theme tags and the set of tuples, respectively.

### *13.5.2 Subscriptions*

Each subscription is a pair of two sets: a set of theme tags and a set of conjunctive attribute-value predicates. Each theme tag is a single word or a multi-word term. Each predicate uses the equality operator to signify exact equality or the tilde operator for approximate equality when indicated. Other Boolean and numeric operators such as $! =$, $>$, and $<$ are kept out of the language for the sake of discourse simplicity. Each predicate consists of an attribute, a value, and specifications of the semantic approximation for the attribute and the value. The most notable feature of the language is the *tilde* $\sim$ operator that helps specify the approximation for an attribute/value when it follows it. An example subscription to energy usage events is as follows:

$$\begin{aligned}&(\{\textbf{power}, \textbf{computers}\},\\ &\{\textbf{type} = \text{increased energy usage event} \sim,\\ &\textbf{device} \sim= \text{laptop} \sim,\\ &\textbf{office} = \text{room 112}\})\end{aligned}$$

The author of the subscription specifies that the device can be a *'laptop',* or something related semantically to *'laptop'*. The subscription also states that the attribute *'device'* itself can be semantically relaxed. However, it states that the event's *'office'* must be exactly *'room 112'*.

The formal definition of the language model is as follows: let $S$ be the set of subscriptions, let $TH$ be the set of all possible theme tags, and let $A$ and $V$ be the sets of possible attributes and values, respectively, which can be used in a subscription. Typically, there are no restrictions on $A$ or $V$, and the user is free to use any term or combination of terms. Each predicate is a quadruple which consists of the attribute, the value, and whether or not the attribute/value is approximated. Let $P$ be the set of possible predicates, thus $P = \{p\text{: } p = (a, v, appa, appv) \in A \times V \times \{0, 1\}^2\}$. A subscription $s \in S$ is a pair $(th, pr)$ where $th \subseteq TH$ and $pr \subseteq P$ are the set of theme tags and the set of predicates, respectively. The degree of approximation is the proportion of relaxed attributes and values. An exact subscription has 0% degree of approximation.

### *13.5.3 Matching*

The matching model is illustrated in Fig. 13.3. An approximate semantic single event matcher $M$ decides on the semantic relevance, or mapping, between a subscription $s$ and an event $e$ based on the semantic mapping between attribute-value predicates of $s$ and attribute-value tuples of $e$. The model is detailed in [151, 272].

An example mapping between the event and the approximate subscription described above is as follows:

$$\begin{aligned}\sigma = \{(&\mathbf{type} = \mathbf{increased\ energy\ consumption\ event}\\ &\leftrightarrow \text{type : increased energy usage event}),\\ (&\mathbf{device} \sim= \mathbf{laptop} \sim\leftrightarrow \text{device : computer}),\\ (&\mathbf{office} = \mathbf{room\ \ 112} \leftrightarrow \text{office : room 112})\}\end{aligned}$$

$M$ works in two modes: the top-1 mode that decides on the most probable mapping between $s$ and $e$, and the top-$k$ mode which decides on the top-$k$ probable mappings to be used later for complex event processing.

The formal definition of matching is as follows: let $C = s \times e$ be the set of all possible correspondences between predicates of $s$ and tuples of $e$. $\forall c = (p, t) \in C \Rightarrow p \in s \wedge t \in e$. $\Sigma = 2^{C}$ is the power set of $C$ and represents all the possible mappings between $s$ and $e$. There are exactly $n$ correspondences in any valid mapping $\sigma$ where $n$ is the number of predicates in the subscription $s$.

For any valid mapping $\sigma$, a probability function quantifies the probability of every predicate-tuple correspondence $(p, t) \in \sigma$ such as (device = laptop$\sim$ $\leftrightarrow$ device: computer). There also exists a probability function that quantifies the probability of the overall mapping $\sigma$, among other possible mappings. Both functions form probability spaces $P_{\sigma}$ and $P$.

In the basic approximate semantic matching model, the semantic relatedness is directly calculated from vector representations of terms as suggested by the distributional semantic model. In the thematic model, probabilities are calculated based on the combined similarity matrix that is based on the thematic pairwise attributes or values semantic relatedness scores. Thematic semantic relatedness measure uses the tags to project and adjust the vector representations of words in a parametric vector space model before calculating their similarity as illustrated in Fig. 13.3 and detailed in [272].

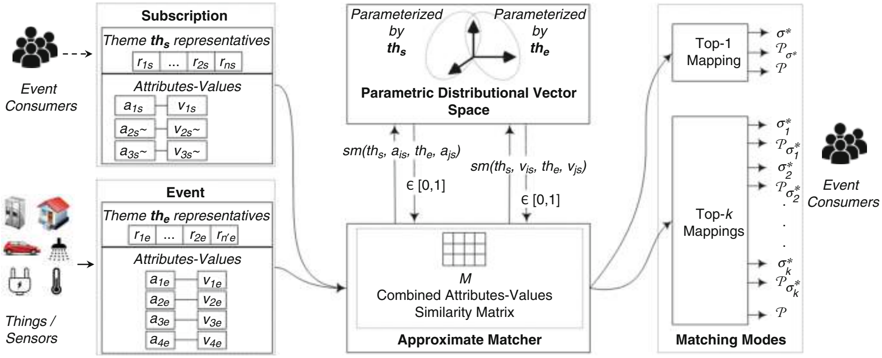


**Fig. 13.3** The matching model (Adapted from [272])

## 13.6 Evaluation and Discussion

To evaluate the models, an evaluation event set of 50,000 events has been semantically expanded out of seed event sets from actual deployments of IoT intelligent systems for smart city, energy management, building, and relevant datasets, to evaluate the approximate semantic event matching model as detailed in [151]. Similarly, 14,743 events were generated to evaluate the thematic event matching model as detailed in [272].

Evaluation metrics can be classified into two categories: effectiveness and efficiency metrics [311]. Effectiveness metrics measure the quality of event matching. A fundamental requirement is the existence of a ground truth which divides events into relevant and irrelevant with respect to each approximate subscription. *Precision*, *Recall*, and combined $F_1Score$ have been used for effectiveness evaluation. The metric used for evaluating time efficiency is the matcher's *Throughput* defined as *Throughput* = (*Number of processed events*)/(*Time unit*).

Additionally, to measure the loosening in the semantic coupling, we use two measures: *alternative number of exact subscription rules* that would be needed in a semantically coupled model, and the *degree of approximation* used in the approximate subscriptions. In the thematic model, the *number of tags* used is also considered. These measures are compared to the exact matching model's numbers, which would typically have many exact subscription rules that have zero degrees of approximation because of the direct coupling.

### *13.6.1 Evaluation of the Approximate Semantic Event Matching Model*

The efficiency evaluation aims to compare the throughput of the approximate semantic matching model against an exact matching model based on query rewriting. Given a set of approximate subscriptions, each approximate subscription can be rewritten as a set of conjunctive statements in the Esper event engine, each of which is a set of attribute-value pairs resulting by replacing the approximate parts of a subscription with related terms from the WordNet dictionary. The ground truth's thesaurus is Merriam-Webster [312].

The experiment was conducted with ten sets of 10−100 approximate subscriptions of 50% degree of approximation using Explicit Semantic Analysis (*ESA*) as a semantic relatedness [313] measure. Figure 13.4 shows that the approximate matcher delivers 94–97% matching quality, which is higher than the 89–92% delivered by the WordNet-based rewriting approach equipped with exact matching. The rewriting approach outperforms the approximate model in throughput when the pairwise semantic relatedness scores are calculated at run-time. However, the approximate matcher based on pre-computed *esa* scores outperforms in throughput with around 91,000 events/s compared to around 19,100 events/s on average.

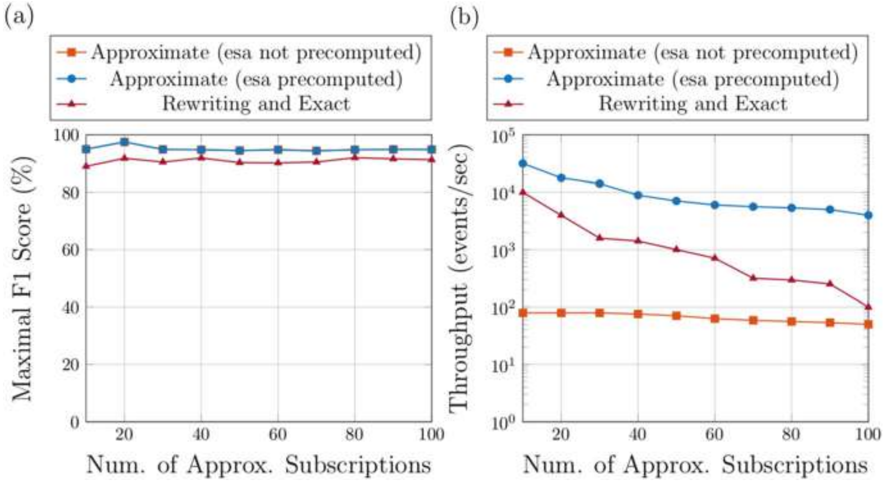


**Fig. 13.4** The approximate semantic event matcher results. (**a**) Effectiveness. (**b**) Efficiency [151]

Finally, to achieve the 100% of $F_1Score$ and the throughput of an exact matcher, there is a need to write manually all the possible rules that are equivalent to the approximate rules. To quantify this situation, we measure how many exact rules are required to compensate for approximate rules given that the rewriting is done with the ground truth thesaurus, which is Merriam-Webster. This showed that about 74,000 exact rules are needed to cover all events compared to a maximum of only 100 rules for the approximate matcher. Thus, the exact matcher is a non-feasible solution in highly semantically heterogeneous environments.

### 13.6.2 Evaluation of the Thematic Event Matching Model

To evaluate the thematic approach, we compare it with non-thematic approximate semantic event processing described above. A large event set is generated with a specific theme as well as a set of subscriptions which assume no semantic agreements and 100% degree of approximation. The thematic matcher is compared with the non-thematic matcher when various theme tags are used.

The baseline matcher achieves 62% of $F_1Score$ and a throughput of 202 events/s averaged over five runs which represent its worst case due to full approximation of the subscription by using the $\sim$ operator on all subscription's predicates.

Each cell in Fig. 13.5 represents the average $F_1Score$ of the sample of five sub-experiments, each of which uses a different combination of events and subscriptions themes tags. For instance, the sub-experiments of the cell in the second column and tenth row from the bottom left, all use two terms to describe the theme of an event, and ten terms to describe a subscriptions theme, and the event theme terms set is a subset of the subscription theme terms set.

Figure 13.5 shows that thematic matching outperforms the non-thematic matching in *F₁Score* for more than 70% of combinations with scores 62–85% and an average of 71% versus 62% for the baseline. Thematic matching performs worse when the number of thematic tags is very small, for example, using just one term as a theme tag. The performance is worst in the bottom triangular half of the figure with F1Score widely ranging from 4% to 62%. Larger themes for subscriptions quickly improve the effectiveness as opposed to the opposite effect by event themes. This reflects the asymmetric relationship between the many heterogeneous events versus fewer subscriptions. Thus, more terms are needed in subscription themes to discriminate relevant events.

Figure 13.6 shows the average throughput for each combination of events and subscriptions theme tags. It suggests that the thematic approach outperforms the non-thematic matcher for more than 92% of the sub-experiments, with a throughput of 202–838 and an average of 320 versus 202 events/s. The improved throughput is due to the thematic filtering of the space during the thematic projection phase, which saves time during the semantic relatedness calculation.

The results show that the thematic approach is limited when users can provide only a small number of tags for subscriptions, and when hard real-time deadlines are required. Otherwise, the results suggest that the use of fewer terms to describe events, around 2–7, and more to describe subscriptions, around 2–15, can achieve a good matching quality and throughput together with low error rates. This is concentrated in the middle to the upper left side of Figs. 13.5 and 13.6. The

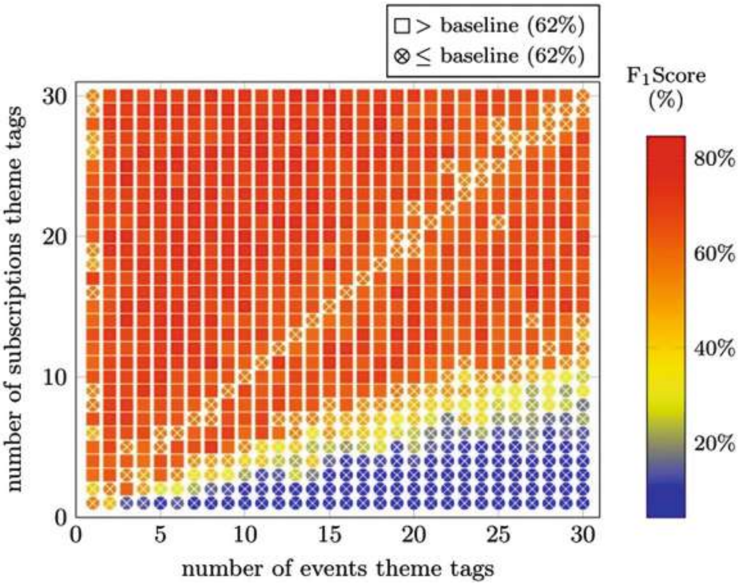


**Fig. 13.5** Effectiveness evaluation of the thematic event matcher [272]

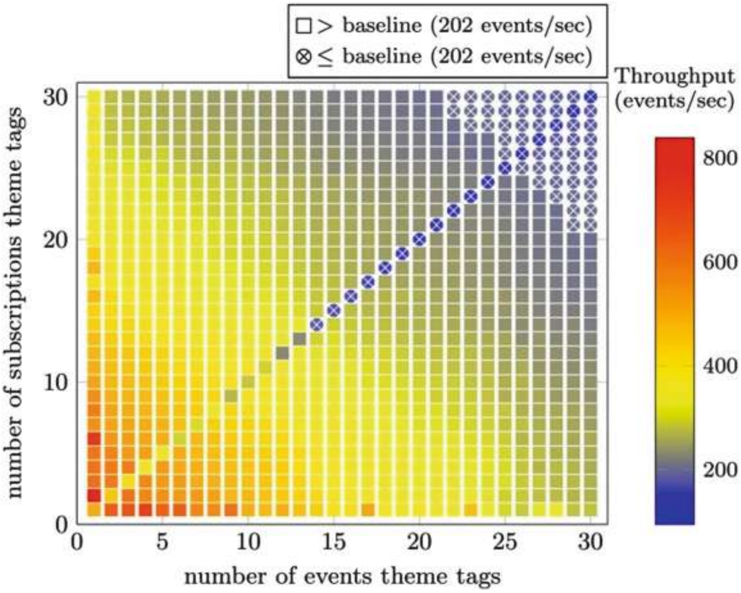


**Fig. 13.6** Efficiency evaluation of the thematic event matcher [272]

evaluation data indicates that events and subscriptions need to be associated with only a few thematic tags.

## 13.7 State-of-the-Art Analysis

The event processing literature related to approximate semantic event matching can be classified into five major classes:

- *Content-Based Event Processing*: In content-based event processing, event sources and consumers use the same event types, attributes, and values without any additional description of meaning external to the rules and events. The principal works in this category are those by Carzaniga et al. [314] (SIENA), Eugster et al. [315], and Fiege et al. [316] (Rebeca). Such approaches are effective with the timely matching and routing of events, but they assume an implicit agreement on the semantics of events outside of the event engine, which is a type of semantic coupling that does not scale in heterogeneous environments.
- *Concept-Based Event Processing*: In this category, participants can use different terms and values and still expect matchers to be able to match them correctly thanks to explicit knowledge representations such as thesauri and ontologies that encode semantic relationships between terms. The principal works in this category are those by Petrovic et al. [317] (S-ToPSS), Wang et al. [318] (OPS), Zeng and Lei [319], and Blair et al. [320] (CONNECT). Given agreements on explicit

models, efficient and effective detection of positive and negative matchings can be achieved. Nonetheless, agreements on explicit models may become an unfeasible task to achieve due to high levels of heterogeneity at large scales.

- *Approximate Event Processing*: Approaches in this category are distinguished by a matching model that is not Boolean. The principal works in this category are those by Zhang and Ye [321] (FOMatch), Liu and Jacobsen [322, 323] (A-TOPSS), and Wasserkrug et al. [324]. These approaches reduce semantic coupling due to their ability to deal with the uncertainties of users about semantics. Time efficiency is high, but effectiveness is lower due to the approximate model, which allows some false-positive/-negatives to occur.
- *Query-Based Fusion*: Approaches in this category adopt declarative languages similar to SQL. These languages support operators of semantics like relational join. They enable semantic description and matching of events as well as the fusion of streams of events with background context data. The principal works in this category are those by Arasu et al. [325] (CQL), Teymourian et al. [326], Le-Phuoc et al. [298] (CQELS), and Anicic et al. [299] (EP-SPARQL). These approaches are like concept-based models in that they assume an explicit model of semantics which might be hard to agree on.
- *Semantic and Context Transformation*: Approaches in this category handle events individually and perform a set of transformations on them to move from one semantic model to another. The principal works in this category are those by Freudenreich et al. [327] (ACTrESS), and Cilia et al. [328, 329] (CREAM). These approaches consider semantics to have one nature and impact on event matching. They are effective and efficient in matching. Nonetheless, semantic models that depend on ontologies and conversion functions require agreements which form a coupled mode that limits scalability in heterogeneous environments.

The literature analysis shows that related approaches are mainly based on symbolic semantics, exact matching, and ad hoc domain specificity, which generally requires agreements that are difficult to achieve in highly distributed and open environments such as smart environments.

## 13.8 Summary and Future Work

This chapter discusses the approximate semantic event processing model in answer to the requirement of loose semantic coupling, which is necessary to adopt the principles of dataspaces to real-time data. We found that to meet this requirement, the event processing paradigm needs to be enhanced with additional elements such as: sub-symbolic distributional semantics, free event tagging, and approximation. We showed that these elements could transform the event matcher into an approximate semantic matcher with a probabilistic model. The resulting approximate and thematic matchers provide an effective matching quality and efficient time performance, and most importantly, they require minimal upfront agreements among event producers and consumers on event semantics.

Future directions would include the development of the approximate matcher to encompass new event and subscription models beyond attribute-value models, and the extension of subscription languages with numeric operators such as less-than and greater-than operators. An interesting direction is the extension of the thematic matching model to other unstructured event types such as images and videos, which would imply opportunities for new intelligent applications in real-time dataspaces of unstructured data [30].

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

# Part IV
# Intelligent Systems and Applications

The fourth part of this book explores the use of Real-time Linked Dataspaces within real-world smart environments by demonstrating its role in enabling intelligent water and energy management systems. This part includes chapters on the development of Internet of Things (IoT)-enabled digital twins and intelligent applications, IoT-enhanced user experience, and autonomic source selection for advanced predictive analytics.

# Chapter 14
# Enabling Intelligent Systems, Applications, and Analytics for Smart Environments Using Real-time Linked Dataspaces

**Keywords** Dataspaces · Intelligent systems · Energy management · Water management · Smart building · Smart airport · Smart home · Smart environments

## 14.1 Introduction

The design of next-generation smart environments poses significant technical challenges with data management, data integration, and real-time processing of dynamic data, and non-technical challenges such as engaging end-users and supporting cultural and organisational changes. Real-time Linked Dataspaces (RLD) are data platforms designed explicitly to tackle these challenges. In this chapter, we provide an overview of how a Real-time Linked Dataspace enables the creation of intelligent systems within several smart environments to demonstrate and evaluate their effectiveness.

This chapter is structured as follows; it begins in Sect. 14.2 with a discussion of the challenges to delivering intelligent energy and water management systems. Section 14.3 introduces Real-time Linked Dataspaces. Section 14.4 details the real-world pilots where the dataspace approach was deployed and evaluated, including an overview of the different user groups involved in each pilot. Section 14.5 details the different types of intelligent applications that were developed using the dataspace, and the chapter concludes in Sect. 14.6.

## 14.2 Intelligent Energy and Water Management

There is a significant opportunity to accelerate the use of intelligent systems to improve energy and water resource management and conservation by analysing, designing, and implementing intelligent systems to increase demand and supply

© The Author(s) 2020
E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_14

efficiency. To manage energy and water holistically within a smart environment, it is essential to use decision support tools that present meaningful and contextual information about usage, price, and availability of energy and water intuitively and interactively to users. Users will need different forms of information to manage their energy and water consumption, from home users managing their personal water usage, business users managing the water consumption of their commercial activities, to municipalities managing regional distribution and consumption at the level of a city or a region. To develop intelligent applications for these diverse users, it is necessary to leverage knowledge from several different domains including metering (household, neighbourhood), water collection and catchment management, energy generation, environmental impacts, water quality, distribution networks, end-user feedback, occupancy patterns, and meteorological data. The design of next-generation intelligent energy and water management systems poses significant technical challenges with data management, data integration, and real-time processing of dynamic data, and non-technical challenges such as engaging end-users and supporting cultural and organisational changes (see Chap. 2). To support the interconnection of intelligent systems in the data ecosystem that surrounds a smart environment, there is a need to enable the sharing of data among systems.

## 14.3 Real-time Linked Dataspaces

A data platform can provide a clear framework to support the sharing of data among a group of intelligent systems within a smart environment [1] (see Chap. 2). In this book, we advocate the use of the dataspace paradigm within the design of data platforms to enable data ecosystems for intelligent systems.

A dataspace is an emerging approach to data management that recognises that in large-scale integration scenarios, involving thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources [2]. Within dataspaces, datasets *co-exist* but are not necessarily fully integrated or homogeneous in their schematics and semantics. Instead, data is integrated on an *as-needed* basis with the labour-intensive aspects of data integration postponed until they are required. Dataspaces reduce the initial effort required to set up data integration by relying on automatic matching and mapping generation techniques. This results in a loosely integrated set of data sources. When tighter semantic integration is required, it can be achieved in an incremental *pay-as-you-go* fashion by detailed mappings among the required data sources.

We have created the Real-time Linked Dataspace (RLD) (see Chap. 4) as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data and real-time stream and event processing capabilities to support a large-scale distributed heterogeneous collection of streams, events, and data sources [4]. To validate the RLD approach, it has been used in the development of intelligent applications and decision support for five smart energy and water environments. The remainder of this chapter details these pilots.

## 14.4 Smart Environment Pilot Deployments

Over the past number of years, we have been involved in a number of projects [18, 62, 63] concerned with investigating the use of a Real-time Linked Dataspace as a data platform for intelligent systems within smart environments, specifically targeting intelligent systems for smart energy and water management. As detailed in Fig. 14.1, the five pilot smart environments are Airport, Office, Home, Mixed Use and School.

### *14.4.1 Smart Airport (Linate, Milan)*

Linate Airport represents large-scale commercial energy and water consumers with mixed water use from washing activities, toilets, restaurants, flight operations, to safety critical infrastructure for emergency response. The intelligent systems and applications in Linate target a variety of users in a business environment (including

| | Airport | Office | Home | Mixed Use | School |
|---|---|---|---|---|---|
| Location | <br>Linate Airport,<br>Milan, Italy | <br>Insight,<br>Galway, Ireland | <br>Houses,<br>Thermi, Greece | <br>Engineering,<br>NUI Galway | <br>Coláiste Na<br>Coiribe, Ireland |
| Target Users | • Corporate users<br>• ~9.5 million passengers<br>• Utilities management<br>• Maintenance staff<br>• Environmental managers | • 130 staff<br>• Office consumers<br>• Operations managers<br>• Utility providers<br>• Building managers | • Domestic consumers (adults, young adults and children)<br>• Utility providers | • Mixed/Public consumers<br>• Building managers<br>• 100 staff<br>• 1000 students (ages 18 to 24) | • Mixed/Public consumers<br>• School management<br>• Maintenance staff<br>• 500 students (ages 12 to 18)<br>• 40 teachers |
| Infrastructure | • Safety critical<br>• 10 km water network<br>• Multiple buildings<br>• Water meters<br>• Energy meters<br>• Legacy systems | • 2190 m$^2$ space<br>• 22 offices + 160 open plan spaces<br>• Conference room<br>• 4 meeting rooms<br>• 3 kitchens<br>• Data centre<br>• 30 person café<br>• Energy meters | • 10 households<br>• Typical variety of domestic settings including kitchen, showers, baths, living room, bedrooms, and garden<br>• Water meters | • Water meters<br>• Energy meters<br>• Rainwater harvesting<br>• Café<br>• Weather station<br>• Wet labs<br>• Showers | Water meters<br>Energy meters<br>Rainwater harvesting |

**Fig. 14.1** Five smart environments [4]

executives, operational managers, and technical staff), each one having a specific role in the company's structure. Apart from the company's employees, applications also target the airport passengers representing the public. Multiple legacy building management systems are in use across the different buildings. The variety of sensors used in the airport require the management of heterogeneous events and their availability to applications in near-real-time. Significant contextual data from the airport's operational systems are needed to process the events for decision-making, presenting significant challenges with legacy data integration and access control.

### *14.4.2 Smart Office (Galway, Ireland)*

The Insight Building hosts approximately 130 staff in a dedicated building with 2190 $m^2$ of space, comprising offices, open plan workspaces, a 90-seat conference room, four meeting rooms, three kitchens, one air-conditioned data centre, and a 30-person café. The building was built in the 1990s and did not employ a building management system or an energy management system. The building has been retrofitted with energy sensors and a simple energy management system. As typically in an organisation, Insight has several information systems that run its operations, including finance and enterprise resource planning, budgeting, and Office IT assets. These enterprise systems can help in identifying energy wastage and promote conservation actions.

### *14.4.3 Smart Homes (Municipality of Thermi, Greece)*

The Municipality of Thermi in Greece provides a residential smart water pilot constituting a representative sample of ten domestic residences with different profiles. The target audience of smart home water applications are the resident adults and children. Other interested users include the municipality management and a developer community for smart home "Apps", research scientists, and the local water utility. Data from Internet of Things (IoT) devices in each home needs to be managed in a near-real-time manner to provide feedback to users on their water consumption. Sensor data needs to be enriched and linked to entities describing the households' water outlets. Securely sharing datasets with both the research and the developer community was an essential requirement.

### *14.4.4 Mixed Use (Galway, Ireland)*

The Engineering Building at NUI Galway is a state-of-the-art smart building with large quantities of sensors and actuators for its management. This smart environment is designed to be a "living laboratory" where the building itself is an interactive teaching tool. The building includes lecture halls, classrooms, offices, laboratory facilities, café, showers, and bathrooms. Intelligent energy and water applications target staff members, managers, technicians, researchers, and students. As with the smart home example, the requirement to make data easily reusable by occupants in the environment is an important requirement. Also, staff members are interested in understanding usage behaviours and detecting saving opportunities, and students are interested in visualising the building consumption and utilising data from the environment in their projects and research works.

### 14.4.5 Smart School (Galway, Ireland)

Coláiste na Coiribe is an Irish language secondary school that has been constructed in 2015 at a suburban location in Galway City in Ireland. It includes classrooms, offices, sports halls, and associated toilet and shower facilities. The school accommodates students aged 12 to 18 years, together with teaching and operational staff. The school has been fitted with a commercial state-of-the-art building management system to manage its energy and water consumption. An essential requirement in this environment is to customise the communication of water and energy data for the diverse range of school stakeholders.

### 14.4.6 Target Users Groups

Smart environments can engage a wide range of end-users with different interests and priorities, from corporate managers looking to improve the performance of their business to school children who want to explore and learn more about sustainability and their effects on the environment. The target users across the five pilot smart environments (as illustrated in Fig. 14.2) are:

- *Building Managers and Operations Staff*: Managers, technicians, and engineers as well as other staff members with responsibility for the operations of buildings. These are adult users that have an advanced level of education and are highly skilled.
- *Passengers*: Passengers that range from business travellers to a variety of casual travellers from different age groups, from kids to adults.
- *Corporate Staff*: Office and administration staff that are found in a typical organisation. These adults are often highly educated and skilled (e.g. administration, HR, marketing, and sales), but not in the technical area of energy and water management.
- *Families*: Family units with potentially multiple generations in one home, including children, young adults, adults, parents, and grandparents.

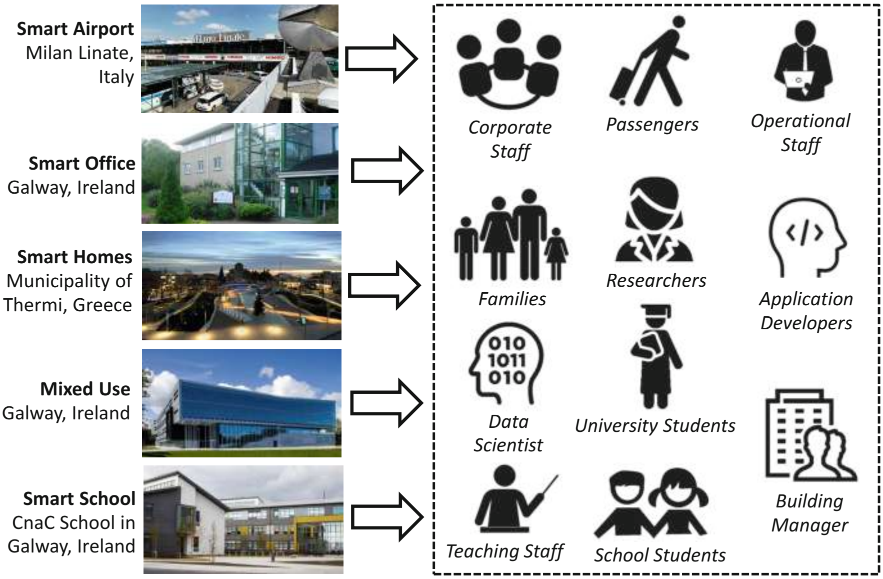


**Fig. 14.2** Target user groups across different pilot sites

- *Teaching Staff*: Teaching staff at both university and school levels looking to use the smart environment as a teaching aid within their classes.
- *University Students*: Undergraduate and research students are interested in understanding their environment and using the data in their projects and research works. The age groups of this group range from young adults to adults.
- *School Students*: School students ranging from kids to young adults.
- *Data Scientists*: Leverage algorithms and machine learning to extract knowledge and insights from data in the smart environment.
- *Researchers*: Advanced users who want to analyse the data from the smart environment within their discipline specific (i.e. architecture, cognative science) research projects.
- *Application Developers*: Software and application developers who want to create intelligent applications (including data analytics) for the users within the pilots.

## 14.5 Enabling Intelligent Systems, Applications, and Analytics for Smart Environments

Within a smart environment, data platforms such as Real-time Linked Dataspaces (RLD) [4] are valuable for application developers and data scientists as they constitute a one-stop shop of all the data required for building their intelligent applications or analytics: open data, enterprise data, and sensor data. RLD include public or private data sources that can be used to drive innovations and new solutions within

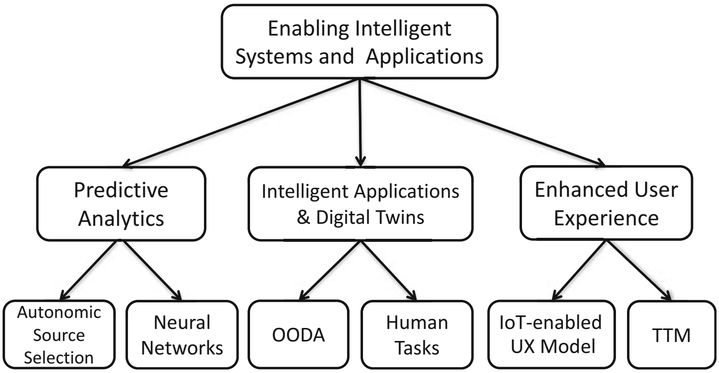

Enabling Intelligent Systems and Applications
Predictive Analytics
Intelligent Applications & Digital Twins
Enhanced User Experience
Autonomic Source Selection
Neural Networks
OODA
Human Tasks
IoT-enabled UX Model
TTM

**Fig. 14.3** Enabling intelligent systems and applications using the Real-time Linked Dataspace

the smart environment. In this part of the book (see Fig. 14.3), we examine the use of the RLD to provide source selection for real-time predictive analytics, personalised applications for users, and enhanced user experiences for intelligent systems within smart environments.

- *Predictive Analytics*: A data platform giving access to IoT and open data sources is particularly promising when creating real-time predictive data analytics for decision support within intelligent systems. The main challenge here is the dynamic selection of sources from the dataspace to support predictive analytics. In this context, we are interested in the study of automatic source selection for prediction models in the energy domain using open data and IoT. Chapter 15 presents an autonomic computing inspired approach to source selection for training deep neural networks and machine learning algorithms within the RLD.
- *Intelligent Applications and Digital Twins*: A key challenge in delivering smart environments is creating effective applications for end-users with new digital infrastructures within the environment. In Chap. 16, we reflect on the experience of using the RLD for developing over 25 different IoT-based intelligent applications and digital twins within five different smart environments, from Airports to Schools. The goal of these has been to engage users within intelligent systems to increase water and energy awareness, management, and conservation. The overall design philosophy of the intelligent applications and digital twins has been guided using Boyd's "OODA Loop" for decision-making.
- *User Experience*: Creating a compelling user experience within a smart environment (from smart buildings to smart cities) is an essential factor to success. In Chap. 17, we reflect on our experience of developing IoT-based intelligent applications using the RLD where the goal has been to engage a wide range of users (from building managers to business travellers) to increase water and energy awareness, management, and conservation. The design of the experience is defined using an IoT enhanced model for user experience. The user journey within the smart environment is supported by leveraging the Transtheoretical Model of behaviour change to influence a person's attitude positively towards sustainable behaviour.

## 14.6 Summary

The design of next-generation smart environments poses significant technical challenges with data management, data integration, and real-time processing of dynamic data, and non-technical challenges such as engaging end-users and supporting cultural and organisational changes. Real-time Linked Dataspaces (RLD) are data platforms designed explicitly to tackle these challenges. In this chapter, we provide an overview of how an RLD enables the creation of intelligent systems within a number of smart environments to demonstrate and evaluate their effectiveness to deliver intelligent energy and water management systems. Real-world pilots for smart office, home, school, and airport were introduced together with the different user groups involved in the pilots. Finally, the role of the RLD in enabling intelligent applications for the smart environment from predictive analytics, personalised applications, to enhanced user experience, is introduced.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

# Chapter 15
# Autonomic Source Selection for Real-time Predictive Analytics Using the Internet of Things and Open Data

Ninad Arabekar, Wassim Derguech, Eanna Burke, and Edward Curry

**Keywords** Source selection · Predictive analytics · Autonomic computing · Decision support · Internet of Things · Open data · Dataspaces

## 15.1 Introduction

Real-time predictive data analytics is a very important tool for effective decision support within intelligent systems. When making decisions using data, it is critical to use the most appropriate data. When creating predictive analytics, the selection of data sources is important as the quality of the sources influences the accuracy of the predictive model. Within a smart environment, a dataspace is valuable for data scientists as it provides a one-stop shop of all the data required for creating their analytical models: enterprise data, Internet of Things (IoT), sensor data, and open data. However, the increase in the number of data sources presents a challenge in selecting the most appropriate data source to use. The co-existence approach of dataspaces results in them containing much more data sources than within traditional data management approaches. This means that the need to perform source selection is an ongoing activity; as the dataspace is incrementally improved, sources will need to be re-examined to determine their suitability for tasks. We propose an autonomic source selection service for predictive analytics for intelligent systems within a smart environment. This service has been evaluated in real-world settings using a Real-time Linked Dataspace for energy predictions using IoT sensor data and open weather data.

The chapter is structured as follows: Discussion in Sect. 15.2 details the challenges of source selection for analytics. Section 15.3 provides an overview of the autonomic source selection approach, including its architecture and the suitability of prediction models. Section 15.4 details the source selection workflow and the criteria for reselection. In Sect. 15.5 we explore the source selection service together with machine learning models in two real-world intelligent systems. The chapter concludes with a summary in Sect. 15.6.

© The Author(s) 2020

E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_15

## 15.2 Source Selection for Analytics in Dataspaces

Real-time data sources are increasingly forming a significant portion of the data generated in smart environments. This in part is due to increased adoption of the Internet of Things (IoT) and the use of sensors for improved data collection and monitoring of daily activities in smart buildings, smart homes, smart cities, and others. In this section, we explore the need for new data support services to deal with the increased number of data sources and the resulting challenge of selecting the most appropriate real-time data source for building predictive models within intelligent systems.

### *15.2.1 Real-time Linked Dataspaces*

To support the interconnection of intelligent systems in the data ecosystem that surrounds a smart environment, there is a need to enable the sharing of data among intelligent systems. A data platform can provide a clear framework to support the sharing of data among a group of intelligent systems within a smart environment [1] (see Chap. 2). In this book, we advocate the use of the dataspace paradigm within the design of data platforms to enable data ecosystems for intelligent systems.

A dataspace is an emerging approach which recognises that in large-scale integration scenarios, involving thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources [2]. Within dataspaces, data sources *co-exist* and are not necessarily fully integrated or homogeneous in their schematics and semantics. Instead, data is integrated on an *as-needed* basis with the labour-intensive aspects of data integration postponed until they are required. Dataspaces reduce the initial effort required to set up data integration by relying on automatic matching and mapping generation techniques. This results in a loosely integrated set of data sources. When tighter semantic integration is required, it can be achieved in an incremental *pay-as-you-go* fashion by detailed mappings among the required data sources.

We have created the Real-time Linked Dataspace (RLD) (see Chap. 4) as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data and real-time stream and event processing capabilities to support a large-scale distributed heterogeneous collection of streams, events, and data sources [4]. In this chapter, we focus on the source selection support service of the RLD. The selection of the correct data source is an important challenge in a dataspace. As the dataspace is incrementally improved, sources will need to be re-examined to determine their suitability for tasks. In Chap. 11, we explored the challenges of selecting event services based on their quality of service. In this chapter, we look at the classic source selection problem for creating predictive models from real-time stream analytics.

### *15.2.2 Internet of Things Source Selection Challenges*

A multitude of Internet-connected devices generating data can quickly become infeasible to cope with. In a traditional data analytics scenario, decisions were driven by insights from information queried over statically stored tabular/relational data. However, with the increasing use of IoT devices, various business domains, governments (e.g. cities), and citizens can unlock the value of low-level data from sensor devices. Much of this data is now available as open data for public use. Choosing the right data source is an important part of effective decision-making within intelligent systems.

Optimal decisions can be made if only the most appropriate data streams are used within the decision-making process and predictive models [31]. However, it is seldom possible to manually decide in advance on the appropriate data sources for a specific application in a real-time big data streaming environment. Once decisions are made using data, it becomes crucial that the best quality data is used. Thus, it is imperative that data-driven decision models are built upon data streams that provide accurate and precise predictions while being tolerant of faults. Data quality issues in data-driven decision-making in critical domains can have disastrous consequences. The importance of source selection can be evident in intelligent systems which involve heavy presence of IoT sensors:

- *Autonomous Vehicles:* Semi/fully autonomous vehicles depend on IoT data streams for emergency roadside assistance, and real-time traffic alerts. The choice of right data streams can help these vehicles decide the best course of action. However, the selection of anomalous speed/direction/proximity sensors could result in accidents.
- *Wind Farm Energy Generation:* Multiple IoT sources from wind turbines, as well as open data sources for weather conditions and forecasts, need to be consulted to build efficient prediction models for wind farms [330]. However, defective sensors monitoring parts of power-generating turbines could lead to failure in maintaining the optimum performance.
- *Building Energy Management:* By leveraging the IoT sensors within a smart building, it is possible to predict the energy use of a smart building based on the weather forecast and the usage patterns of the building [25]. However, an error in the temperature monitoring system of a building could lead to wastage of fuel required for heating.

Another aspect that compounds the source selection dilemma for intelligent systems is the dynamic nature of data sources within an IoT-based smart environment. For example, given the task of real-time predictive modelling over high-velocity data streams, source selection is required to be quick, responsive, and autonomous in behaviour.

The problem of data source selection within a dataspace essentially boils down to identifying the most appropriate data streams that can be harnessed to build useful data models for descriptive as well as predictive analytics. The accuracy of these predictive models forms the basis of selection criteria for the underlying data stream sources. Thus, a suitable approach is expected to be efficient and effective in the following aspects of the source selection problem:

- *Accuracy:* Use of machine learning models to achieve high accuracy.
- *Low Maintenance:* Source selection is autonomous, robust, and fault tolerant.
- *Highly Scalable:* Ability to withstand fluctuations in the number of data sources, data volume, or velocity.
- *Enrich Quality Metadata:* Update the quality of service of a data source, based on its performance for productive tasks.

## 15.3 Autonomic Source Selection Service for Real-time Predictive Analytics

The selection of data sources is important as the data from the sources influences the results of predictive analytics. In order to design our source selection service, we studied the available literature. In a 2011 review of trust in networked datasets [331], the authors noted that the process of selecting a data source is subjective based on the needs of the consumer. A conventional method for selecting a dataset to answer a query is to examine the metadata associated with the data source, for example, size of the dataset, date and frequency of updates [332]. Another method for determining correct information is to establish a consensus from several sources [331].

The co-existence approach of dataspaces results in them containing much more data sources than within traditional data management approaches. This means that the need to perform source selection is an ongoing activity; as the dataspace is incrementally improved, sources will need to be re-examined to determine their suitability for tasks. This constant change in dataspaces can be accompanied by rapid changes in data quality, which in turn affects their predictive power. Within the context of IoT, the scale of the data has increased, and for real-time predictive analytics, it is imperative that source selection should occur with minimum manual intervention. In order to meet these requirements, the source selection service of the RLD leverages techniques from autonomic computing to make the process as independent and self-managed as possible.

### *15.3.1 Autonomic Source Selection*

Autonomic computing systems are being developed to cope with large and increasingly complex systems. Autonomic systems can manage themselves when given high-level objectives from administrators by freeing them from low-level tasks [333]. The idea is to reduce the system operation and maintenance time to the minimum possible and allow the system to run at the best of its abilities. The four

**Table 15.1** Four pillars of an autonomic system [333]

| Trait | Explanation |
|---|---|
| Self-configuring | An autonomic application/system should be able to configure and reconfigure itself under varying and unpredictable conditions. |
| Self-optimising | An autonomic application/system should be able to detect suboptimal behaviours and optimise itself to improve its execution. |
| Self-healing | An autonomic application/system should be able to detect and recover from potential problems and continue to function smoothly. |
| Self-protecting | An autonomic application/system should be capable of detecting and protecting its resources from both internal and external attack and maintaining overall system security and integrity. |

pillars of an autonomic system (see Table 15.1) are self-configuration, self-optimisation, self-healing, and self-protection [334].

The selection service is designed to follow the principles of autonomic systems. The design of the selection service supports three of the four autonomic principles (self-protecting is not supported). The approach selects the data source by evaluating the results of the predictive analytics to determine the data source with the best results [335]. The approach maintains and improves the quality of the predictions over time while being self-managing [334]:

- *Self-configuration:*
  - *Automatic installation and initiation:* The source selection service can be installed into any prediction approach, and it automatically starts being useful with minimal intervention by a skilled worker.
  - *Generalisable:* There should be a low configuration effort to adapt the service to another prediction model to encourage re-use.
- *Self-optimisation:*
  - *Select the best data sources:* The service chooses the best sources of data to make the best possible predictions.
  - *Adapt to changes in the operation of the environments:* The predictions should react to changes in the operational phase, for example, expansion or contraction of a workforce or extensions/renovations to a building. Thus, source selection needs to adapt to operational changes.
  - *Low user interaction:* The service should continue working with no supervision.
- *Self-healing:*
  - *Transparent failover of a data source*: In the case of a failure of a data source (e.g. a weather station malfunction), the service should continue to make the best-effort prediction using an alternative data source so that agents dependent on it can continue to operate.

- *Maintain high-quality predictions*: Predictions must remain accurate as poor predictions may cause consumers of the data to make wrong decisions.
- *Timely identification of faults:* Faulty data sources (e.g. a damaged sensor) should be identified quickly so that an alternative data source can be used.

### *15.3.2 Architecture*

The autonomic source selection service is designed according to the architecture depicted in Fig. 15.1. The service is part of the support services within the RLD, which is used to support the management of the data sources. The autonomic service is designed following the MAPE-K control loop from IBM [336] that consists of stages for Monitoring, Analyses, Planning, and Execution, all sharing a common Knowledge base.

Within the source selection service, these stages perform the following activities:

- *Monitor:* The monitor samples the outputs of the predictive models and stores the prediction for later comparison with the actual values. It is also responsible for

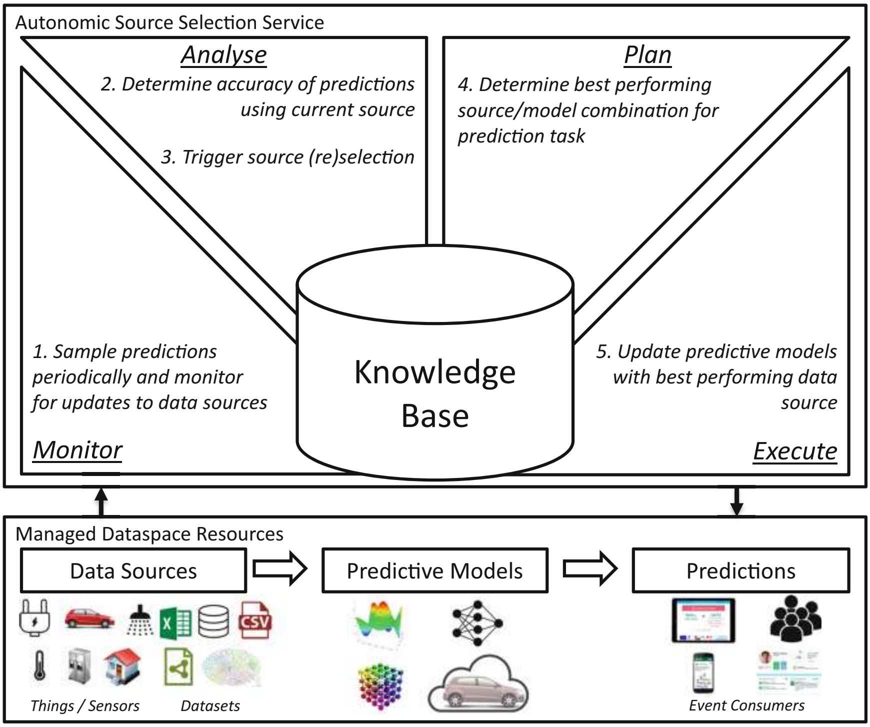

Autonomic Source Selection Service
Analyse
2. Determine accuracy of predictions using current source
3. Trigger source (re)selection
Plan
4. Determine best performing source/model combination for prediction task
Knowledge Base
1. Sample predictions periodically and monitor for updates to data sources
Monitor
5. Update predictive models with best performing data source
Execute
Managed Dataspace Resources
Data Sources
Predictive Models
Predictions
Things / Sensors
Datasets
Event Consumers

Fig. 15.1 Autonomic source selection service following the MAPE-K control loop

observing any changes to the sources in the dataspace, such as new sources joining or updates to existing sources.

- *Analyse:* The main objective of this stage is to compare the predictions from the models with the actual readings to generate an error percentage, which is then used to determine the quality of the predictions. We use the Mean Absolute Percentage Error (MAPE) and Root Mean Squared Error (RMSE) as error indicators. The analysis is run at regular intervals to determine how well the models are performing.
- *Plan:* The planning stage is responsible for deciding when to select the sources used for prediction. Planning involves building many predictive models, which is costly and time-consuming, so trade-offs should be made between the frequency of reselections and maintaining the best prediction model possible. We suggest using hard limits of 15 min and 1 month for the upper and lower bounds of the reselect interval, but the specific interval should be kept dynamic. This planning activity builds a prediction model from the best available sources in the RLD.
- *Execute:* Updates the sources and prediction models that are using the source selection service within the dataspaces.
- *Knowledge Base:* The knowledge base is used to store and share data (e.g. source/ model performance, and error rates) between the different stages in the MAPE-K.

### 15.3.3 Prediction Models

We identified a set of requirements for choosing the right machine learning algorithm for the prediction models [335]. These requirements are listed below in descending order, from highest to lowest priority:

- *Accuracy:* The model should generate accurate predictions.
- *Fast Model Generation:* The model should be quickly generated.
- *Efficient with Minimal Data*: The model should be able to be deployed and quickly make accurate predictions. This requires the service to not overfit to the training set and result in drastically incorrect predictions.
- *Supports Nominal and Numeric Inputs*: Both nominal and numeric data will be used as inputs and need to be handled by the prediction model.
- *Generalisation Outside of the Available Training Data:* This is important as the prediction model will be used in a real-time scenario where data encountered will often be outside the range of data in the training set.
- *Low Configuration:* Minimal configurations effort required for a portable service.
- *Low Pre-processing of Data:* Pre-processing of the data does not require a skilled user. This is often automated when using a software suite.
- *Insights into Factors Influencing Prediction:* Dependency analysis is generated for user information and understanding.

**Table 15.2** Comparison of machine learning algorithms [335]

| | Multiple linear regression | Artificial neural network | Regression tree | Kernel regression analysis | Support vector machines |
|---|---|---|---|---|---|
| Insight into input importance | Yes | No (sensitivity analysis possible [337]) | Yes (tree shows which variables are important) | Yes | Yes |
| Overfitting prevention | Not prone to over fitting | Methods available | Pruning to stop overfitting may be required | Not prone to overfitting | Not prone to overfitting |
| Ease of implementation | Simple | Requires manual tuning of nodes and layers | Simple to understand and implement | Moderate | Moderate—optimisation exist |
| Computational cost | Low | Typically, high. Depends on training function | Low | Depends on training function | High on large data, scales O($n^2$) to O($n^3$) (adaptations available [338]) |
| Other benefits | Simple and quick | De facto solution for regression on non-linear data<br>Extensive literature | Simple and quick | Works well outside of training data range | Works well outside of training data |
| Disadvantages | Poor with non-linear relationships | Prediction outside of training data can be drastically incorrect (corrections exist for this)<br>Unimportant inputs may worsen predictions | Predictions not in continuous range-binned values | Needs normalising of input data | Needs normalising of input data |

Concerning these requirements, we carried out a comparison of the common machine learning algorithms found in the literature in order to understand their utility. The comparison is shown in Table 15.2.

The source selector needs to be scalable and efficient in its operation. The technical challenges this service faces are memory use, processing time, and latency. As the service generates multiple prediction models using training sets that potentially can span a considerable time, it is required to drop references to datasets as soon as they are not needed, so garbage collection to free memory can take place.

The processing time should be kept low by selecting sources with a greedy-type method. Evaluation of data sources is initially done over a large number of sources with small datasets, and it changes to a more comprehensive evaluation for fewer sources. As such, effort spent on poor data sources is reduced. Latency (from queries to a remote data store) is addressed by only querying the source for data that is necessary and by reducing duplicate queries where possible.

## 15.4 Autonomic Source Selection Workflow

The workflow for autonomic source selection service has two definitive stages: (1) initial model training with historical data, and (2) evaluation with real-time data streams. Both stages play an essential role in ensuring that the best data source from the dataspace is utilised for efficiently performing predictive model over real-time data streams.

**Stage 1: Initial Model Training with Historical Data** To forecast with real-time data streams, it is crucial that these models are tuned finely for different data sources as well as predictive algorithms are used. This stage aims at learning the optimum value of hyperparameters using various combinations of data streams and machine learning algorithms. The training happens with a significant number of historical observations accumulated over a considerable period. A large sample size helps in reducing bias (overfitting) while training the predictive models.

**Stage 2: Evaluation with Real-time Data Streams** Once the different predictive models are trained with optimum hyperparameters over multiple data streams, the same models are used for forecasting by using the real-time data feed. The models mostly try to predict the dependent values as close as possible to the observed ones. So, in the case of a classification problem, the F1 score would be the primary metric of evaluation. Also, specificity, recall, precision, or sensitivity might be considered depending on the type of classification task at hand. In the case of a regression task, the models are evaluated based on minimising the prediction error quantified with Root Mean Square Error (RMSE) and variance score. The data sources are then considered in increasing order of their RMSE scores (for regression) or decreasing order of F1 scores (for classification). The top-performing data source model is selected dynamically for predicting the outcome of real-time data streams.

### *15.4.1 4-Step Workflow*

Figure 15.2 is a step-wise representation of the autonomic source selection methodology. The essence of the approach is involving historical observations in learning about the quality of data source qualities and using machine learning to define their predictive power. These steps are elaborated below:

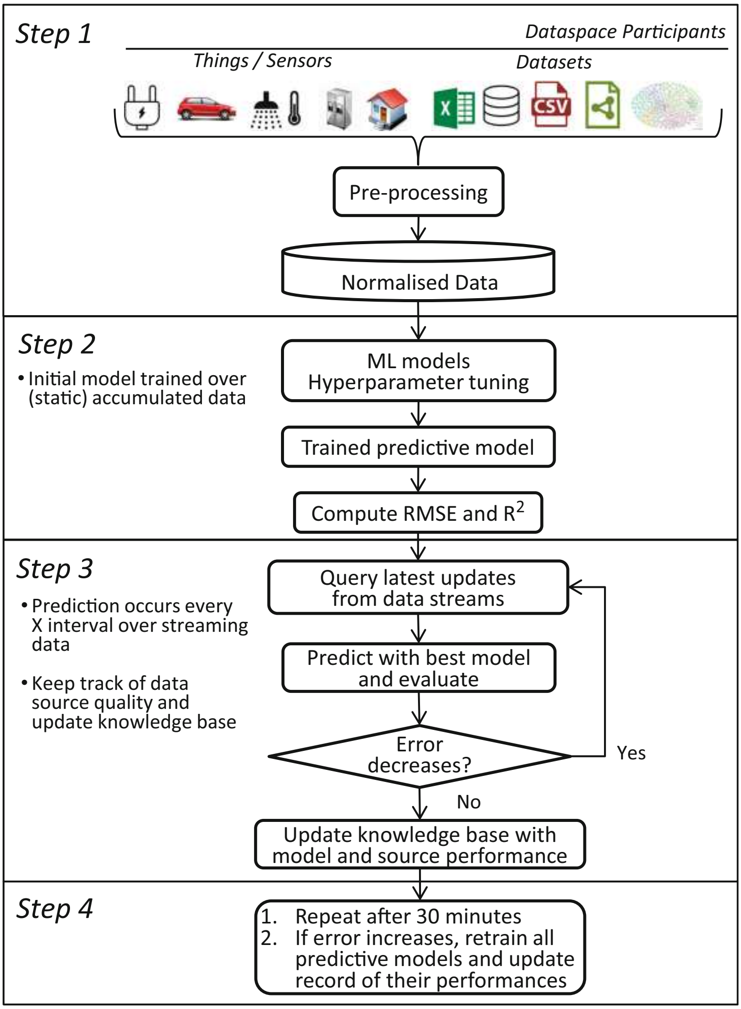

Step 1
Dataspace Participants
Things / Sensors
Datasets
CSV
Pre-processing
Normalised Data
Step 2
• Initial model trained over (static) accumulated data
ML models
Hyperparameter tuning
Trained predictive model
Compute RMSE and R2
Step 3
• Prediction occurs every X interval over streaming data
• Keep track of data source quality and update knowledge base
Query latest updates from data streams
Predict with best model and evaluate
Error decreases?
Yes
No
Update knowledge base with model and source performance
Step 4
1. Repeat after 30 minutes
2. If error increases, retrain all predictive models and update record of their performances

**Fig. 15.2** Workflow for autonomic source selection

- *Step 1—Autonomous Data Procurement and Pre-processing:* Attributes from the different sources (e.g. wind speed, wind direction, temperature) within the RLD are accumulated from different data streams for a period (e.g. 24 h). Since data sources and streams can serve data in different formats, an ETL (extract, load, transform) pipeline may need to be set up to collect and standardise it autonomously (see Chap. 6 for further details on normalising data sources in the RLD). The next part of the pre-processing is to aggregate and normalise the data. Data collected from multiple sources during the same period needs to be merged and used as features. Any data generated in the same duration from the prediction target would need to be included here to serve as dependent variables explained by the features. This consolidated data would be aggregated on an interval basis to evaluate the predictive power of different sources during different times of the day. Since sources could return data reflecting similar conditions in the smart environment, collinearity between different features would be observed. This would help in detecting multi-collinearity issues in regression modelling of the prediction target. Also, relevant features (e.g. wind speed/direction/temperature) from different sources would be identified.

- *Step 2—Model Training (Hyperparameter tuning for ML models):* The aggregated data is now split in a 90:10 ratio with the 10% allocated for testing. The predictive models are trained, tested, and cross-validated to obtain different values of RMSE with an array of hyperparameter values for the Neural Network. Hyperparameters for which the lowest RMSE is observed after tenfold stratified cross-validation would be retained for predictive modelling with real-time streaming data.
- *Step 3—Model Evaluation (Iterative model evaluation with subsets of data sources):* RMSE and Mean Absolute Error (MAE) are calculated as a mean of relevant values obtained from cross-validation over the trained model. A subset of the data source with the least RMSE and MAE are retained, and others discarded. In subsequent iterations, other data sources are considered along with best performing one, to check if the error is minimised further. The adjusted coefficient of determination (Adj. $R^2$) which measures explained variance (while penalising the addition of new explanatory variables) would be taken into account. New data sources will be considered if RMSE and MAE are decreasing and adj. $R^2$ increases. The subset of data streams that minimise the error while maximising the explained variance is chosen as an optimum set of data sources initially. The performance evaluation of the models and data sources is stored in the knowledge base for future analysis.
- *Step 4—Dynamic Source (Re)selection:* It is possible that data streams return erroneous data or fail at a certain point in time. To counter such a situation, the service would build predictive models over a suitable time window for the prediction timeframe (e.g. 30 min for energy predictions) to check the RMSE and Adj. $R^2$ scores. In case the performance dips below a certain threshold, other data streams would be considered for building the model. If the performance metrics show improvement with new data sources, the original malfunctioning data stream would be replaced.

### 15.4.2 Reselection Triggers

The criteria for the reselection of a source from the dataspace are shown in Table 15.3 and discussed further below.

**Table 15.3** Autonomic source reselection criteria [335]

| Reselection trigger | Reasoning | Mechanism | Autonomic characteristic |
|---|---|---|---|
| Timed error checks. | Checking if the prediction model has become less accurate. | Query the internal error knowledge base for the sources being used. | Self-optimising<br>Self-healing. |
| Timed builds. (regardless of errors) | Data collected may allow a more accurate prediction model than the current one. | Send flag to reselect. | Self-optimising. |
| Very high error in the incoming stream. | May indicate a failure in a sensor or data source. | Short-term error check. | Self-healing. |
| New source event received. | New source may be more accurate than existing sources. | Send flag to reselect. | Self-configuring<br>Self-optimising. |

**Timed Error Check** This is done by checking the recent performance against the expected performance of the prediction model. This check is for long-term trends in the dataset. Error checks are less costly than builds, so they happen on a more frequent basis than timed builds. The error check becomes more frequent every time the error returned is too high and less frequent when the error found is under the acceptable threshold. Changeable conditions make it check more often to include the newest and most relevant data and prevent data ageing. The error check is for 25% of the time the prediction model is in place so that the newest data is given high priority.

The error check uses Student's t-distribution to determine when the mean-error is too high. It does this by checking that the average error is lower than a threshold computed using: Threshold = (expectedMAPE) + 0.674 $*$ (standard deviation).

During the evaluation of source selection service, we found that this equation corresponds to a 75% one-tailed test, that is, 75% of predictions should be lower than this error %. The 0.674 figure is valid for more than 120 data instances, which corresponds to 30 h of data. For example, if the mean-error is 7.819% and the standard deviation of the error is 6.411%, the threshold would be: Threshold = (7.819) + 0.674$*$(6.411) = 12.14.

**Timed Build** This is for the initial implementation phase of the service. It addresses the potential for improvement of the prediction model, whereas the error checking prevents degradation of the quality of predictions. The time interval begins at 15 min and increases by 50% every time a reselect flag is sent due to the time interval being exceeded. The time interval between reselections is reset if a reselect message is sent due to an error check or a new source detected.

**High Error Detected** It checks for deteriorations in the quality of the predictions such as a failure of a data source or any other error. This uses the previous formula, with 10% of the time the prediction model in place: Threshold = (expected MAPE) + 2$*$(standard deviation).

**New Source Detected** This is activated if a listener picks up an observation from a new source. This adds the source to the pool of available options sooner than otherwise waiting for the reselect cycle, and it would be beneficial if a new dataset were added to the triple store in bulk with the addition of a new source.

## 15.5 Evaluation Within Intelligent Systems

The autonomic source selection service has been evaluated within two different real-world intelligent systems in the energy domain: (1) Wind Farm Energy Prediction, and (2) Building Energy Use Prediction. Both intelligent systems involved building predictive models using both IoT streams and managing open data within a Real-time Linked Dataspace.

### *15.5.1 Wind Farm Energy Prediction (Belgium)*

Wind power forecasting methods can be used to plan unit commitment, scheduling, and dispatch, and maximise profit by electricity traders [339]. We experiment with the utility of our source selection approach by selecting optimum data streams from multiple weather sources near the wind farms to predict the power generated. Predicting wind power while selecting the best from multiple weather data sources is an interesting challenge. The main set of features determining the wind energy generation is the weather conditions prevailing in the surrounding region. This useful information can be obtained from the data streams from weather stations in the vicinity. However, given the transient nature of this data, relying only on a single source can be potentially detrimental for consistently forecasting highly accurate power values.

The availability of real-time open streams on power generated from wind farms posed an initial problem. However, “Elia”, one of the key electricity transmission system operators in Belgium, does a commendable job of publishing wind energy data frequently throughout the day via REST services as well as downloadable CSV extracts. We considered the weather updates released by the stations located in a 10 km radius range of this wind farm. As seen in Fig. 15.3, we have Elia-connected offshore wind farms (A) and four weather stations (B–E) located in the nearby coastal towns of Ostend, Zeebrugge, Middelkerke, and Knokke Heist. We performed a set of experiments with the source selection service choosing the best weather station to predict the output of the wind farm.

The results of experiments with different periodic prediction windows are summarised in Table 15.4. Although the trends in generated power seem to match perfectly with the wind speed recorded about 6 h ago, it is observed that the 1 h prediction window is the best indicator of power that would be generated from wind.

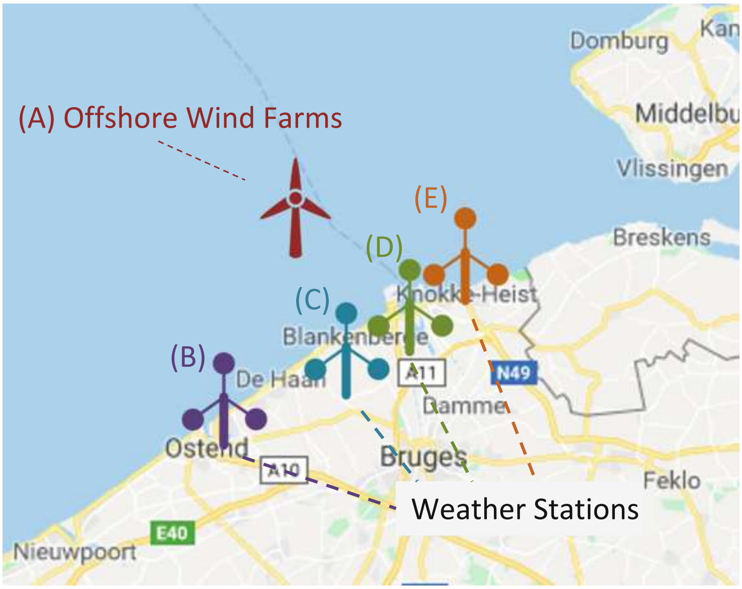


**Fig. 15.3** Locations of weather stations and wind farms

**Table 15.4** Source selection with periodic windows

| Metric | 1 Hour | 2 Hours | 4 Hours | 6 Hours |
|---|---|---|---|---|
| RMSE | 177.15 | 195.84 | 207.7 | 228.54 |
| Variance score | 0.66 | 0.57 | 0.52 | 0.44 |
| Correlation (Wind speed ~ power) | 0.74 | 0.72 | 0.66 | 0.60 |

Using the 1 h window size, multiple models are trained with combinations of relevant machine learning algorithms and weather data sources. Upon training the models, they are evaluated with RMSE and variance score metrics. The sources being transient are trained with algorithms that work well with regression tasks once the models are initially fitted with data; that is, the models fitted with an entire year of data are stored, and when a new weather observation is available on the data stream, they are used to predict the power that would be generated.

The best performing models are highlighted in Table 15.5. The performance is determined based on a low RMSE and a high variance score. The results achieved prove the effectiveness of autonomic source selection service. However, there are some limitations to the approach. For example, our experiments dealt with small volumes of low-latency weather streams. In this intelligent system, the serial training and testing process for the predictive models did not pose any significant performance issues. However, we envisage some performance degradation with high-velocity streams. Also, the approach relies on the data source being described in the catalog with the necessary metadata for their autonomous discovery. However, the source metadata may not pre-exist in some cases, and they would need to be created.

**Table 15.5** Summary of predictive model performance over 1-year data

| | Weather data sources (Weather stations locations) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | Zeebrugge (data source 1) | | Middelkerke (data source 2) | | Ostend (data source 3) | | Knokke (data source 4) | |
| Algorithm | RMSE | Var. | RMSE | Var. | RMSE | Var. | RMSE | Var. |
| Linear regression | 181.58 | 0.65 | **185.79** | **0.63** | 181.18 | 0.64 | 220.92 | 0.47 |
| Support vector machine | 183.58 | 0.64 | 196.17 | 0.59 | 186.09 | 0.62 | 196.87 | 0.58 |
| Artificial neural network | **158.57** | **0.73** | 202.94 | 0.56 | 168.39 | 0.69 | 223.73 | 0.46 |
| Decision tree | 182.72 | 0.64 | 206.45 | 0.54 | 175.83 | 0.66 | 210.19 | 0.52 |
| Ensemble (GBR) | 180.41 | 0.65 | 229.23 | 0.44 | **168.03** | **0.69** | **184.45** | **0.63** |
| Ensemble (AdaBoost) | 187.51 | 0.62 | 217.73 | 0.49 | 211.71 | 0.50 | 218.22 | 0.49 |

### 15.5.2 Building Energy Prediction (Galway, Ireland)

The second intelligent system is at the Smart Building pilot at the Insight Centre at NUI Galway, Ireland. The building has been retrofitted with energy sensors to monitor the consumption of power within the building, including the consumption of devices, light, and heating. All the information from the building is managed within an RLD [100].

The first experiment investigates the accuracy of the service after the initial installation, that is, with no historical weather or electrical power data. Errors in the predictions versus the actual power readings are observed over time (errors for the 3, 6, and 12 hour-ahead predictions). Four machine learning algorithms in WEKA [340] were tested for short (1 week) and long-term datasets (5 weeks). The datasets were for the building's main incoming power and the weather observations of the NUI Galway weather station. For both datasets, the training set comprised of 66% of the available data and the remainder was used as the test set for evaluation. The datasets used for the testing contained the same attributes as the implemented model, that is, 15 min averages of power reading, time, the day of the week, temperature, pressure, humidity, wind speed, and wind direction. The datasets were randomised, and each experiment was performed three times, with average values taken. Unless stated, all configurations (see Table 15.6) are the defaults chosen by developers of the WEKA library ver. 3.7.3.

The results of the evaluation of the machine learning algorithm using short- and long-term datasets are shown in Tables 15.7 and 15.8, respectively. We notice from these tables that the Neural Networks, though more accurate, were far slower than the other two learning algorithms, and were not used in the service. The Linear Regression and Sequential Minimal Optimisation Regression (SMOReg) took approximately the same amount of time, with Linear Regression being more accurate. This may be due to the homogeneity of the summer dataset not presenting non-linear trends. For the implemented service, SMOReg was chosen due to the documented ability to handle non-linear data outside of the training set well, despite the inferior results from testing. The Artificial Neural Networks (ANNs) were

**Table 15.6** Algorithm configurations in WEKA for testing [335]

| Config number | Configuration settings |
|---|---|
| Config 1 | SMOReg. The WEKA implementation of a support vector machine for regression |
| Config 2 | One hidden layer backpropagation ANN with default WEKA values |
| Config 3 | Two hidden layer backpropagation ANN with default WEKA hidden layer one and ten nodes in hidden layer two |
| Config 4 | Linear regression. Default WEKA implementation for multiple linear regression |

**Table 15.7** Machine learning testing results for 1 week [Dataset size = 672, training data size = 443, test data size = 229] [335]

| | Mean absolute error (kW) | RMSE (kW) | Time (s) | Correlation coefficient |
|---|---|---|---|---|
| Config 1 (SMOReg) | 5.3638 | 6.9759 | 0.851 | 0.6233 |
| Config 2 (1 Layer ANN) | 2.7606 | 3.6242 | 47.004 | 0.9158 |
| Config 3 (2 Layer ANN) | 3.0473 | 4.1506 | 50.842 | 0.8961 |
| Config 4 (Linear Regression) | 4.8586 | 5.9283 | 0.759 | 0.7396 |

**Table 15.8** Machine learning testing results for 5 weeks [Dataset size = 3298, training data size = 2176, test data size = 1122] [335]

| | Mean absolute error (kW) | RMSE (kW) | Time (s) | Correlation coefficient |
|---|---|---|---|---|
| Config 1 (SMOReg) | 4.6755 | 6.5965 | 32.4 | 0.8054 |
| Config 2 (1 Layer ANN) | 3.3332 | 4.5841 | 229.7 | 0.9162 |
| Config 3 (2 Layer ANN) | 3.7566 | 4.7279 | 247.0 | 0.9221 |
| Config 4 (Linear Regression) | 4.7579 | 6.0173 | 2.4 | 0.8396 |

initially experimented on hourly power and weather readings for three months from October to December, where different combinations of hidden nodes were tested. During this testing, the second hidden layer with ten nodes was found to improve the RMSE by 1% over the single layer ANN. This improvement did not carry over to the more granular data in the real service, where adding the second hidden layer decreased the accuracy of the service.

## 15.6 Summary

In this chapter, we detail an autonomic source selection service for a dataspace to support the evaluation of real-time data streams for predictive analytics. The source selection service is designed using the principles of an autonomic system to reduce the administrative overhead. The service was developed and tested on two real-world intelligent applications in the energy domain. The evaluation shows that the service was effective in supporting the choice of source and machine learning technique most appropriate to build predictive models in the energy domain.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

# Chapter 16
# Building Internet of Things-Enabled Digital Twins and Intelligent Applications Using a Real-time Linked Dataspace

**Edward Curry, Wassim Derguech, Souleiman Hasan, Christos Kouroupetroglou, Umair ul Hassan, and Willem Fabritius**

**Keywords** Decision support · Internet of Things · Intelligent systems · Smart environments · Energy management · Water management · Dataspaces

## 16.1 Introduction

Smart environments have emerged in the form of smart cities, smart buildings, smart energy, smart water, and smart mobility. A key challenge in delivering smart environments is creating intelligent applications for end-users using the new digital infrastructures within the environment. In this chapter, we reflect on the experience of developing Internet of Things-based digital twins and intelligent applications within five different smart environments from an airport to a school. The goal has been to engage users within Internet of Things (IoT)-enabled smart environments to increase water and energy awareness, management, and conservation. The chapter covers the role of a Real-time Linked Dataspace to enable the creation of digital twins, and an evaluation of intelligent applications.

The chapter starts in Sect. 16.2 with a description of Digital Twins and the role that Boyd's OODA Loop can play in their realisation. Creating digital twins and intelligent application using a Real-time Linked Dataspace is detailed in Sect. 16.3. The results from the smart energy and water pilots are detailed in Sect. 16.4. Section 16.5 discusses experiences and lessons learnt, and the chapter concludes in Sect. 16.6.

© The Author(s) 2020

E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_16

## 16.2 Digital Twins and Intelligent Applications with a Real-time Linked Dataspace

Driven by the adoption of the Internet of Things (IoT), smart environments are enabling data-driven intelligent systems that are transforming our everyday world, from the digitisation of traditional infrastructure (smart energy, water, and mobility), the revolution of industrial sectors (smart autonomous cyber-physical systems, autonomous vehicles, and Industry 4.0), to changes in how our society operates (smart government and cities). To support the interconnection of intelligent systems in the data ecosystem that surrounds a smart environment, there is a need to enable the sharing of data among systems.

### *16.2.1 Real-time Linked Dataspaces*

A data platform can provide a clear framework to support the sharing of data among a group of intelligent systems within a smart environment [1] (see Chap. 2). In this book, we advocate the use of the dataspace paradigm within the design of data platforms to enable data ecosystems for intelligent systems.

A dataspace is an emerging approach to data management which recognises that in large-scale integration scenarios, involving thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources [2]. Within dataspaces, datasets *co-exist* but are not necessarily fully integrated or homogeneous in their schematics and semantics. Instead, data is integrated on an *as-needed* basis with the labour-intensive aspects of data integration postponed until they are required. Dataspaces reduce the initial effort required to set up data integration by relying on automatic matching and mapping generation techniques. This results in a loosely integrated set of data sources. When tighter semantic integration is required, it can be achieved in an incremental *pay-as-you-go* fashion by detailed mappings among the required data sources.

We have created the Real-time Linked Dataspace (RLD) (see Chap. 4) as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data, knowledge graphs, and real-time stream and event processing capabilities to support a large-scale distributed heterogeneous collection of streams, events, and data sources [4].

### *16.2.2 Digital Twins*

Within the business community [32], the metaphor of a “Digital Twin” is gaining popularity as a way to explain the potential of IoT-based assets and smart environments. A digital twin refers to a digital replica of physical assets (car), processes

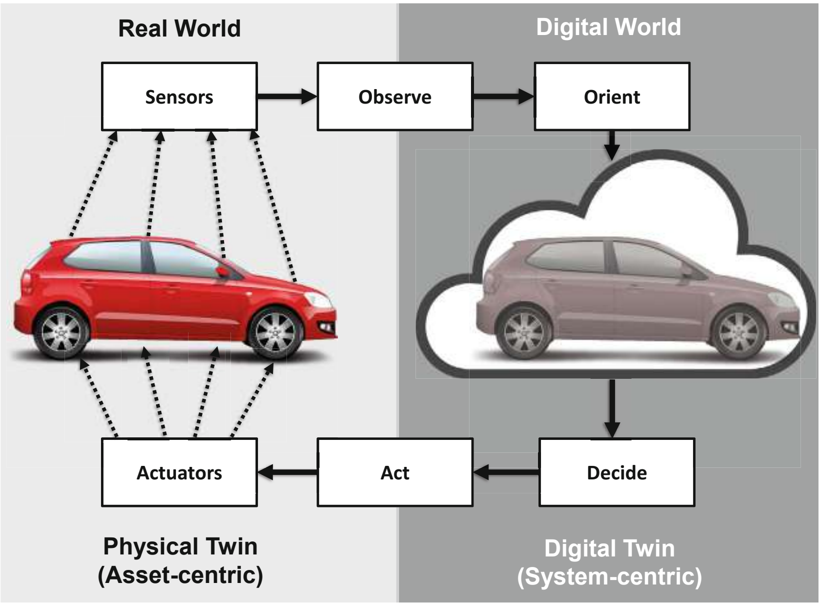


**Fig. 16.1** A digital twin provides a digital representation which can be analysed to optimise the operation of the "physical twin"

(value-chain), system, or physical environment (building). The digital representation provided by the digital twin can be analysed to optimise the operation of the "physical twin". The digital twin provides a digital representation (Fig. 16.1) (i.e. simulation model, data-driven model) that updates and changes as the physical twin changes. Digital twins can provide digital representations ranging from human organs such as the heart and lungs to aircraft engines and city-scale twins. For example, the SmartSantander smart city project has deployed tens of thousands of Internet-connected sensor devices in large cities across Europe [33]. The sensing capabilities of these devices are wide-ranging, including solar radiation, wind speed and direction, temperature, water flow, noise, traffic, public transport, rainfall, parking, and others. The devices provide a digital representation of the state of the real world, in the case of SmartSantander a digital representation of the city, enabling visibility into processes and operations of the city that can be analysed and optimised.

With the use of advanced analytics and artificial intelligence techniques, the digital twin can learn the optimal operating conditions of the physical twin and optimise the physical twins' operations in areas such as performance, maintenance, and user experience. One of the most promising outputs from such an analysis is the possibility to find root-causes of potential anomalies which can happen (prediction) and improve the physical process (innovation).

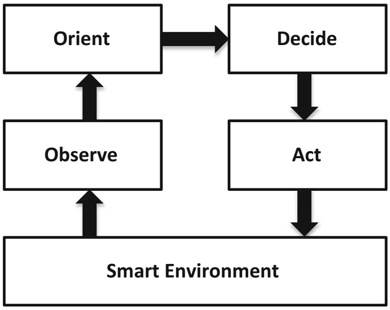
Digital twins are a sophisticated example of a cyber-physical system which is constructed from multiple sources of data including real-time IoT sensors, historical sensor data, traditional information systems, and human-in-the-loop input from human operators and domain and industrial experts. The core of a digital twin requires a holistic and systematic approach to data management and decision-making; at the heart of a digital twin is an *OODA Loop*.

### *16.2.3 The OODA Loop*

John Boyd hypothesised that individuals and organisations undergo a continuous cycle of interaction with their environment. Boyd developed the "OODA Loop" [341] as a decision process by which an entity (either an individual or an organisation) reacts to an event by breaking the decision cycle down to four interrelated and overlapping processes through which one cycles continuously: *Observe*, *Orient*, *Decide*, and *Act* (OODA). Boyd initially applied the OODA Loop to military operations, and it was later applied to enterprise operations. More recently, it has been considered as an approach for processing observations within cyber-physical systems [14]. In this latter context, we apply the OODA Loop as a high-level design guide for intelligent energy and water systems within smart environments. As illustrated in Fig. 16.2, the four OODA processes applied to an intelligent application within a smart environment are:

- *Observation:* The gathering of data from the smart environment to understand its state.
- *Orientation:* The analysis and synthesis of data to form an assessment of the circumstances within the smart environment. Moving from data to information, knowledge, and insights.
- *Decision:* Consideration of the options to determine an appropriate course of action. The goal is to optimise the operation of the smart environment. The use of predictive modelling can play a significant role here.
- *Action:* The physical execution of decisions via actuation (both automated and human). Once the result of the action is observed, the loop starts over.

**Fig. 16.2** Boyd's OODA Loop [341] applied to intelligent applications within a smart environment



## 16.3 Enabling OODA for Digital Twins and Intelligent Applications

As shown in Fig. 16.3, and detailed in [4], we use the OODA Loop to align the different development phases of digital twins and intelligent applications with the relevant RLD support services.

### 16.3.1 Observation

The RLD support services facilitate the observation phase by minimising the required effort for a data source to join the RLD. Support services such as the Catalog, Access Control Service (see Chap. 6), and the Search and Query Service (Chap. 10) are the primary services that enable the collection of data sources and IoT data and the maintainability of its associated metadata. The incremental approach of the RLD made it easier to gradually improve the collection of observations from the smart environment by adding a new sensor, thing, or dataset to the RLD. The 5 star

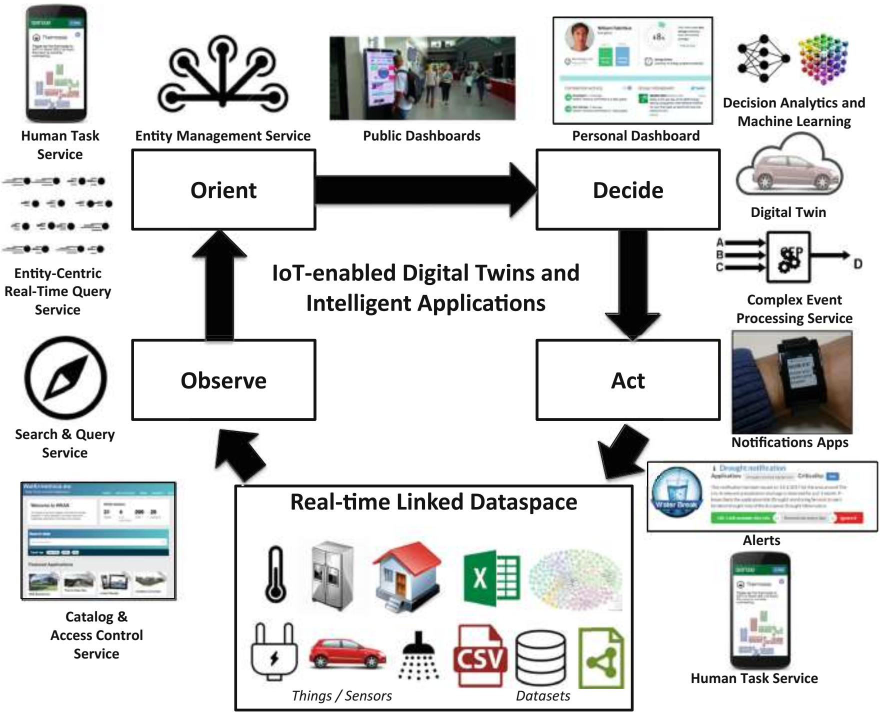

Human Task Service
Entity Management Service
Public Dashboards
Personal Dashboard
Decision Analytics and Machine Learning
Orient
Decide
Digital Twin
Entity-Centric Real-Time Query Service
IoT-enabled Digital Twins and Intelligent Applications
A
B
C
D
Complex Event Processing Service
Observe
Act
Search & Query Service
Notifications Apps
Real-time Linked Dataspace
Alerts
Catalog & Access Control Service
Human Task Service
Things / Sensors
Datasets

Fig. 16.3 Role of RLD and its support services across the phases of the OODA Loop [4]

pay-as-you-go model for data management (see Chap. 4 for more details) was useful for specifying and planning the level of service needed for each data source.

The human task service enables the engagement of users in maintaining a high-quality catalog of managed entities. Active participation of users in a smart environment improves their engagement and sense of ownership while supporting a higher accuracy of data maintained by the dataspace. In one of our pilot deployments we noticed a direct benefit of using the human task service for the collaborative management of the entities in the environment to provide a more accurate and rich understanding of the environment's state [256].

### *16.3.2 Orientation*

The primary objective of the orientation phase is to support situational awareness of the smart environment. The real-time query services (see Chap. 10) enable users to understand the current and historical state of the smart environment. The Entity Management Service (EMS) builds awareness regarding the entities in the environment through entity linking and enrichment (which can be supported by the Human Task Service). Together with the real-time query services, the EMS provides entity-centric views of the smart environment and reduces the overall effort to integrate entity data from different real-time streams and contextual data sources.

Within all the pilots, a key goal is to increase the visibility, understanding, and awareness of energy and water use. Using the RLD support services, we can build dashboards to provide situational awareness for users with targeted information on energy and water consumption. Within the different pilots, this is manifested in a variety of ways and at different time frames, from informing the residents in their smart home as they live their day, supporting the detailed analysis required by building managers and operational staff, to brief encounters with "frequent-flyer" passengers as they pass through the airport. User orientation in the pilots was driven by public displays, interactive touchscreen displays and tablet applications (see Fig. 16.4). These user interfaces communicate current and historical energy and water usage within the environment, convey information about the importance of energy and water, tips on how to improve consumption, and games to calculate the users' footprint in real-time. The displays are also personalised to target different users by using appropriate metaphors to communicate relevant messages to them. The intelligent applications in the orientation phase make extensive use of real-time, historical, and contextual data sources to enhance the user experience (see Chap. 17).

### *16.3.3 Decision*

Once users have built a certain level of awareness regarding the energy or water consumption of their environment, they can use their expertise to start taking

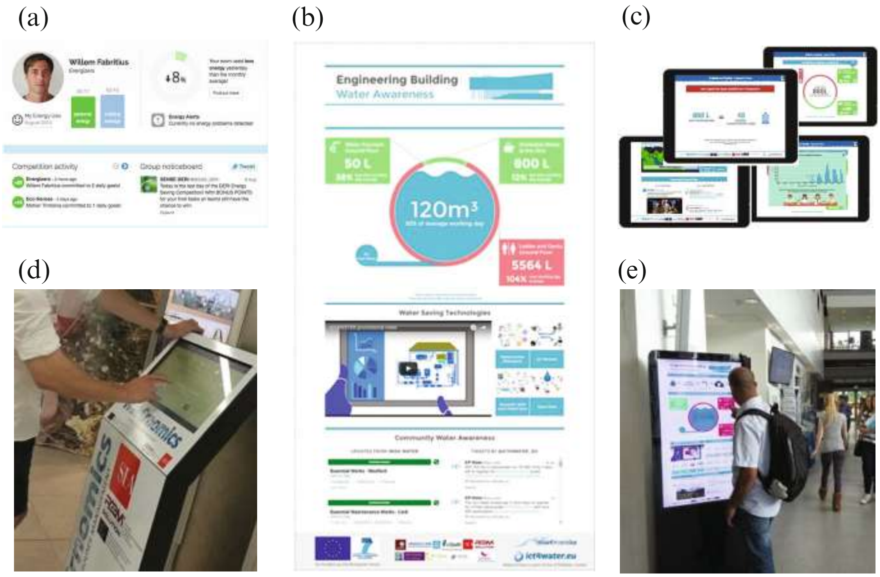
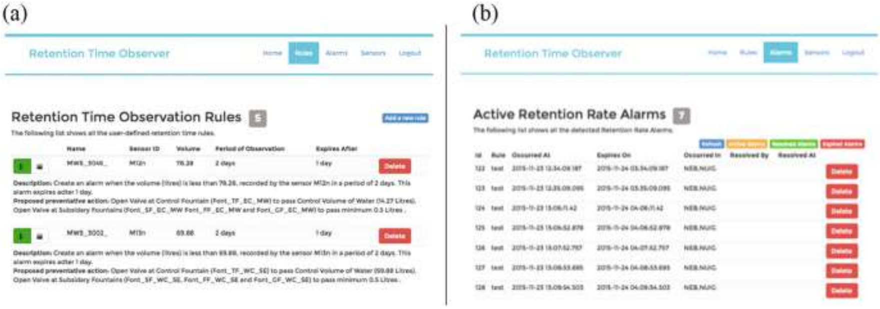


**Fig. 16.4** Public interactive displays and personalised dashboards: (**a**) Smart office, (**b**) Smart building, (**c**) Smart school, (**d**) Smart airport, (**e**) Smart building

decisions towards more sustainable behaviours. In the decision phase, a critical aspect of the dashboards is to provide users with targeted information on usage, goal setting, targets for conservation, and tips to improve their consumption behaviour. This is where decision-making takes place. For example, managers can define consumption thresholds to serve as sources of "alerts", notifying them of excessive usage, goals attained, or the detection of a possible fault (e.g. Complex Event Processing Service, see Chap. 11). Developing decision support applications relies on the entity-centric real-time query service to analyse data from the environment, interpret it, and decide on the appropriate course of action.

A specific example of decision support is the Water Retention Time Observer application (see Fig. 16.5) that determines the amount of time drinking water resides in water pipes and creates alerts in case of potential issues. In public spaces, drinking water quality is a significant concern for building managers: is the water safe to drink? Currently, this can be managed by selecting a popular location to place the drinking water fountains to ensure people are always using them, thus ensuring that freshwater is always flowing through the pipes. However, in some public buildings, drinking water fountains can remain unused during long holidays and weekends. Consequently, drinking water can reside for extended periods in the pipes. In this context, the water retention time observer can assist building managers by providing timely notifications regarding low water quality in drinking water pipes. This is achieved by creating a simple digital twin of the water network to detect inactivity in specific measurement points in the water network and sending a notification if



Retention Time Observation Rules

The following list shows all the user-defined retention time rules.

| | Name | Sensor ID | Volume | Period of Observation | Expires After | |
|---|---|---|---|---|---|---|
| | MWS_3046_ | M12n | 78.28 | 2 days | 1 day | Delete |

Description: Create an alarm when the volume (litres) is less than 78.28, recorded by the sensor M12n in a period of 2 days. This alarm expires after 1 day.
Proposed preventative action: Open Valve at Control Fountain (Font_TF_EC_MW) to pass Control Volume of Water (14.27 Litres). Open Valve at Subsidiary Fountains (Font_SF_EC_MW Font_FF_EC_MW and Font_GF_EC_MW) to pass minimum 0.5 Litres .

| | Name | Sensor ID | Volume | Period of Observation | Expires After | |
|---|---|---|---|---|---|---|
| | MWS_3002_ | M13n | 69.88 | 2 days | 1 day | Delete |

Description: Create an alarm when the volume (litres) is less than 69.88, recorded by the sensor M13n in a period of 2 days. This alarm expires after 1 day.
Proposed preventative action: Open Valve at Control Fountain (Font_TF_WC_SE) to pass Control Volume of Water (59.88 Litres). Open Valve at Subsidiary Fountains (Font_SF_WC_SE, Font_FF_WC_SE and Font_GF_WC_SE) to pass minimum 0.5 Litres .

Active Retention Rate Alarms

The following list shows all the detected Retention Rate Alarms.

| Id | Rule | Occurred At | Expires On | Occurred In | Resolved By | Resolved At | |
|---|---|---|---|---|---|---|---|
| 122 | test | 2015-11-23 12:34:08.187 | 2015-11-24 03:34:08.187 | NEB.NUIG | | | Delete |
| 123 | test | 2015-11-23 12:35:09.095 | 2015-11-24 03:35:09.095 | NEB.NUIG | | | Delete |
| 124 | test | 2015-11-23 13:06:11.42 | 2015-11-24 04:06:11.42 | NEB.NUIG | | | Delete |
| 125 | test | 2015-11-23 13:06:52.878 | 2015-11-24 04:06:52.878 | NEB.NUIG | | | Delete |
| 126 | test | 2015-11-23 13:07:52.757 | 2015-11-24 04:07:52.757 | NEB.NUIG | | | Delete |
| 127 | test | 2015-11-23 13:08:53.695 | 2015-11-24 04:08:53.695 | NEB.NUIG | | | Delete |
| 128 | test | 2015-11-23 13:09:54.503 | 2015-11-24 04:09:54.503 | NEB.NUIG | | | Delete |

**Fig. 16.5** Water retention time observer: (**a**) Observation rules. (**b**) Active alarms

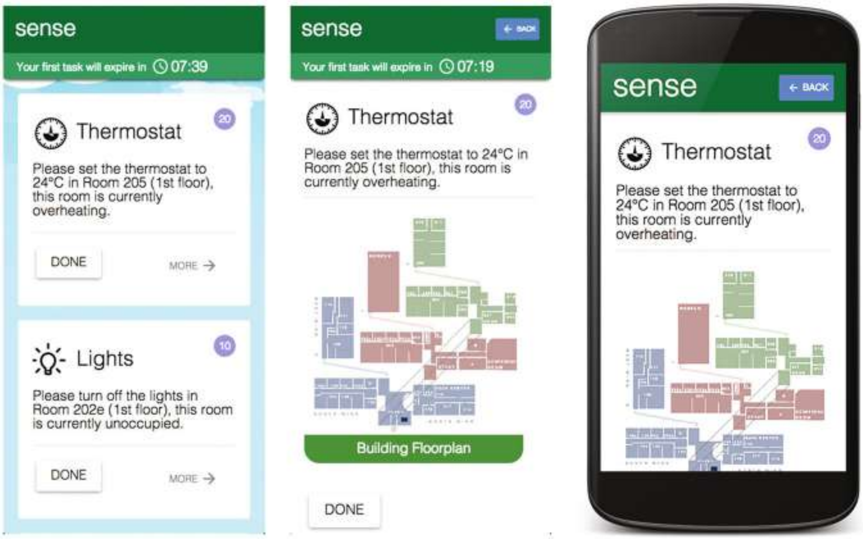
stagnant water is detected. Within this digital twin, we aimed to enable notification to attract users' attention only when necessary. This was a key lesson from our work to enhance user experience, which is discussed further in Chap. 17.

## 16.3.4 Action

Intelligent applications in the action phase of the OODA Loop help users in smart environments meet their goals for energy and water consumption by taking appropriate actions. The complex event processing service of the RLD is used to express these goals as a set of rules which can generate alerts and suggested preventive actions. Actions are then communicated to the users in the smart environment using an appropriate mean of communication: emails, notifications on the dashboards, messages on smart devices, and human tasks.

The occupants of the environment can participate in taking energy or water saving actions. In the smart building pilot, we implemented a collective energy management system where the RLD was used for the identification of energy-saving tasks. The tasks were routed to the building occupants using the human task service (see Chap. 9) to take energy conservation actions such as turning off the light in empty rooms or closing a window when an air conditioner is on in a room. Figure 16.6 shows an example of these "Citizen" actuation tasks.

The role of a building manager is a demanding one that often has personnel working in the field. An anytime-anywhere notification mechanism was needed for managers. To minimise the search friction between actionable information and users, a well-designed notifications system is needed. The wearable info-centre application was developed to enable notification through the wearable technology for high-priority alerts. Figure 16.7 shows an example notification using the wearable info-centre.



sense

Your first task will expire in 07:39

Thermostat 20

Please set the thermostat to 24°C in Room 205 (1st floor), this room is currently overheating.

DONE MORE →

Lights 10

Please turn off the lights in Room 202e (1st floor), this room is currently unoccupied.

DONE MORE →

sense ← BACK

Your first task will expire in 07:19

Thermostat 20

Please set the thermostat to 24°C in Room 205 (1st floor), this room is currently overheating.

Building Floorplan

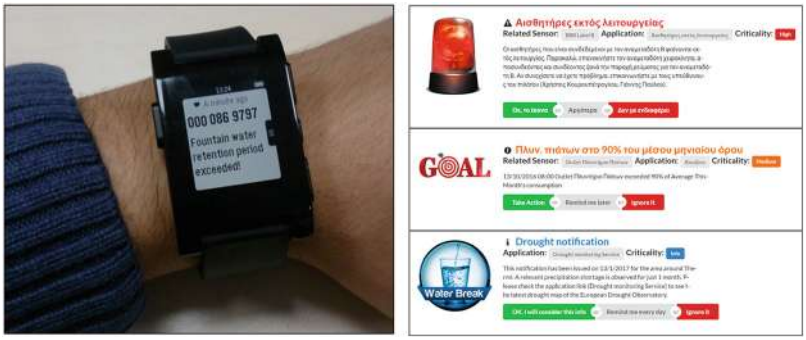
DONE

sense ← BACK

Thermostat 20

Please set the thermostat to 24°C in Room 205 (1st floor), this room is currently overheating.

Fig. 16.6 Example of citizen actuation tasks within the smart environment



000 086 9797

Fountain water retention period exceeded!

Αισθητήρες εκτός λειτουργίας

Related Sensor: Application: Criticality:

Πλυν. πιάτων στο 90% του μέσου μηνιαίου όρου

Related Sensor: Application: Criticality:

Take Action Remind me later Ignore it

Drought notification

Application: Criticality:

OK, I will consider this info Remind me every day Ignore it

Fig. 16.7 Example notifications within the smart environment

## 16.4 Smart Energy and Water Pilots

This section presents the results and insights gained from deploying the RLD and intelligent applications in the smart environments described in Chap. 14. Each pilot followed a similar methodology for design, deployment, and evaluation [63]. In this section, we detail the energy and water savings achieved in the pilots, the performance of the human task services in engaging users to save energy, and a set of experiences and lessons learnt from deploying the RLD in the pilots.

### 16.4.1 Energy and Water Savings

During the initial period of the pilots, energy and water metering data was collected from existing monitoring systems to establish baselines for consumption across the pilots. During the control period, the users within the pilots had access to the data generated by the metering infrastructure system through traditional information systems (e.g. building management systems, and basic public dashboards within the airport, office building, and school). The data collection period for each pilot spanned between 6 and 16 months, which also included a range of user interventions such as pre-surveys, focus groups, interviews, and feedback cycles. The RLD was used to develop intelligent energy and water systems and decision support analytics across the pilot smart environments. Table 16.1 details the characteristics of the pilots during the study period, the number of events generated in the environment, the number of intelligent applications/twins deployed, and savings achieved in terms of energy and water. In terms of energy and water savings, the RLD supports these impacts in three fundamental ways:

- Connecting data across silos provided "big picture" entity-centric views of the resource consumption within the smart environments. These views made it easier for the users within the smart environments (e.g. building managers) to identify waste and efficiency opportunities as the data produced within the environment was structured and organised around real-world entities. Entity-centric views were the basis of the digital twins created.
- The pay-as-you-go approach was useful for building the business case and getting "buy-in" from users by enabling quick wins to demonstrate the benefit of the approach. These early wins that demonstrated energy and water savings encouraged non-technical business users to engage with the project and system more

**Table 16.1** Summary of the impact of intelligent energy and water systems in smart environments [4]

| Pilot site | Location | Study period | Events per year | Intelligent applications/ twins deployed | Actual savings measured | Estimated annual savings |
|---|---|---|---|---|---|---|
| Smart Airport | Linate Airport, Italy | 10 months | ~11.5 million | 8 | 2954 $m^3$<br>3013 kg $CO_2$ | 54,000 $m^3$<br>55,080 kg $CO_2$ |
| Smart Office | Insight, Ireland | 6 months | ~8 million | 4 | 24% energy reduction | – |
| Smart Homes | Thermi, Greece | 16 months | ~2.3 million | 11 | 30% water reduction | – |
| Mixed Use | Engineering Building, NUI Galway | 16 months | ~36 million | 8 | 174 $m^3$<br>177 kg $CO_2$ | 8089 $m^3$<br>8251 kg $CO_2$ |
| Smart School | Coláiste na Coiribe, Ireland | 12 months | ~1 million | 5 | 2179 $m^3$<br>2223 kg $CO_2$ | 9306 $m^3$<br>9492 kg $CO_2$ |

actively. The project team could build a business case around intelligent applications/digital twins and decision support tools that would reduce resource usage and its associated economic costs. The savings identified can be used to justify the necessary investment needed in data integration.

- The RLD enabled highly specialised decision analytics and digital twins that provided action and notification alerts for each of the pilot smart environments, including leak detection, fault detection, and abnormal usage patterns. These alerts and notifications were crucial for building managers and operational staff who do not have time to study and analyse the data generated in the smart environment.

## 16.4.2 Human Task Service Evaluation

To determine the effectiveness of the human task service of the RLD within a smart environment, we performed two types of human tasks in the smart office pilot: (1) an entity enrichment data management task, and (2) a citizen actuation task for energy savings.

### 16.4.2.1 Human Task for Entity Enrichment

This experiment focuses on a data management task that requires the user to enrich the description of an entity by collecting location information on sensors within the smart environment. Accurate location data is needed by the energy management system to make appropriate recommendations about temperature control and energy usage in the monitored building. We do not assume this metadata on sensor locations, and room characteristics are available at the start of the experiment. This situation simulates the case when it is difficult to gather all metadata upfront, or the metadata becomes invalid due to changes in the environment. The objective of the experiment was to use human tasks to enrich the sensor entities in the RLD with the support of building occupants.

The occupants of the building were contacted through email to participate in the experiment. If they consented, they were asked to look for sensors around them in the building and to scan a QR code on the sensors using their mobile phones. This would resolve the URL associated with the QR code in a web browser, where they would then be asked to perform a relevant task. This action connects the user to the human task service within the RLD and enables the linkage between human tasks and physical sensors. Once the participant submits the location of the sensor, additional tasks are pushed to them to collect further metadata about the surrounding environment. Three tasks collect information about lights, heaters, and windows in the room. The collected data is then used to enrich the description of the sensor and room entities in the EMS.

The evaluation is based on the comparison of occupant-contributed metadata versus gold-standard data. The gold-standard data was created manually by studying

**Table 16.2** Description and results of entity enrichment tasks against the gold-standard data [256]

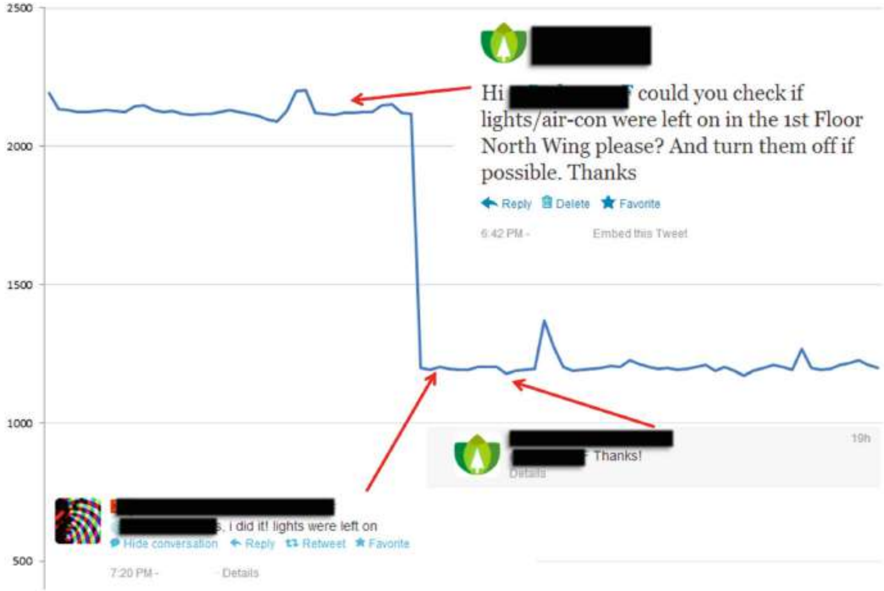
| Task | Description | Assignment method | Accuracy |
|---|---|---|---|
| Sensor location | This task requires participants to specify the location of the sensor. | Task pull based on the QR code. | 85.71% |
| Room lights | This task asks participants to specify the number of fluorescent lights installed in the room. | Task push based on person location. | 100% |
| Room heaters | This task asks participants to specify the number of heaters in the room. | Task push based on person location. | 83.33% |
| Room windows | This task asks participants to specify the number of windows in the room. | Task push based on person location. | 100% |

the physical space. Table 16.2 shows the accuracy of data submitted by occupants of the building within 5 h of sending the invitation email to the building occupants. The reported accuracy is based on the data submitted by the first few participants for each sensor and room. The human task service achieved more than 80% accuracy in describing the sensors and rooms within 5 h. The accuracy could be increased if the results from multiple users are used to verify the accuracy of the contributions.

#### 16.4.2.2 Human Actuation for Energy Savings

The second evaluation of the human task service focuses on tasks for humans to save energy by performing citizen actuation [265]. When the energy management system detects an abnormal energy usage in a room in the building (i.e. high energy use for both the time of day and room status [booked for a meeting or not]), a notification via Twitter is sent to an appropriate user to request the user to check on the issue. This is the actuation request. Often the cause of the energy consumption abnormality is due to a light or equipment (e.g. projector or air conditioning) being left on in an empty room. This interaction between the user and the human task service, together with the relevant energy sensor readings, is illustrated in Fig. 16.8.

Within the smart office pilot, we collected data over a 32-week control period. Weekend data was removed from the experiment, as the users would not be on site. Fifteen volunteers were selected for the experiment. For each request, one volunteer was chosen at random to receive the request. The results of the experiment are illustrated in Fig. 16.9 with the max, min, median, and average energy consumption for the control and actuation days. Overall, the results show that the energy usage on average declined compared to the control during the weeks (experimental weeks) users received actuation requests and completed the actions of turning off electrical components. The average saving was 0.503 kWh, when compared to the average energy used in the control weeks of 1.93 kWh. This equates to a decrease in energy usage by 26%. Each actuation week's energy usage was equal to or lower in value to the lowest control week apart from 1 week (which, compared to the other control weeks, was lower).



[redacted]

Hi [redacted] could you check if lights/air-con were left on in the 1st Floor North Wing please? And turn them off if possible. Thanks

6:42 PM - Embed this Tweet

[redacted] 19h

[redacted] Thanks!

[redacted]

[redacted]s, i did it! lights were left on

7:20 PM - Details

**Fig. 16.8** Citizen actuation task using the human task service of the RLD [220]

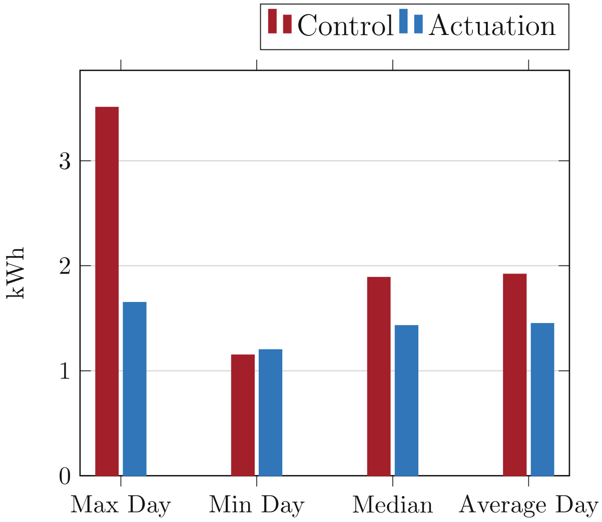


**Fig. 16.9** Daily energy usage—the average, max, min, and median of the control period and actuation period [220]

## 16.5 Experiences and Lessons Learnt

Based on a reflection of our experience of using the RLD in the pilot environments, the following lessons were identified as key learnings to inform the design of future digital twins and intelligent applications for smart environments using the RLD [4].

**Developer Education** Across the pilots, we worked with a diverse set of development teams with different backgrounds, from embedded devices to web front-ends. The dataspace concept was new to most of them, and they were accustomed to working in an environment where they have full-control with the expectation of exact results. Also, the store-and-query culture is more common among developers and users. The processing of data on-the-fly and detecting only data that is of interest in real time, without storage in most cases, can be challenging (aka. event processing) for some developers to understand. Embracing the dataspace took time and required us to demonstrate both the benefits and limitations of the paradigm. Developer education was critical to the adoption of the dataspace. Workshops and tutorials held at pilot sites proved to be an effective mechanism of engaging developers to educate them on the capabilities of the platform and the dataspace data management approach.

**Incremental Data Management Can Support Agile Software Development** The project teams for each pilot operated using an agile software development methodology. The incremental approach of the dataspace and the use of the event-based paradigm were efficient during the design and development phase. The RLD enabled the teams to work at a pace suitable to the stakeholders and data owners involved. The RLD allowed the project team to include new data sources during a development iteration, or to increase the level of integration of an existing source. The decoupling achieved via the catalog and the use of events and streams removed dependencies between parties. It enabled the project teams to work with participants in the pilots in an incremental manner where we could quickly demonstrate value with a low upfront investment in data integration. As the pilots progressed, more and more data became available in the RLD enabling the creation of sophisticated data-intensive intelligent applications, digital twins, and analytics.

**Build the Business Case for Data-Driven Innovation** It is important to clearly articulate the business case for the RLD to justify the necessary investment in data infrastructure. Within our pilots, we discovered a strong business case for data-driven innovation by justifying the investment based on the resulting cost savings achieved due to improving resource efficiency (e.g. energy and water savings). A key challenge was to bring together the different stakeholders in the pilots to support and deliver the project. For example, the IT organisations had the data, but the savings resulting from the system benefit the operations teams of the organisations (e.g. water and energy). Thus, operations have a clear motivation to invest, but IT does not. By bringing these stakeholders together, we were able to build a holistic business case.

**Integration with Legacy Data Is a Significant Cost in Smart Environments** While sensors and connected devices are an essential source of data in a smart environment, they are not the only source of data necessary to make an environment "smart". In our pilots, a considerable number of different legacy data sources needed to be integrated to collect the information necessary to make informed and intelligent decisions. While the RLD provided an effective incremental approach that integrates legacy data at a minimum cost, it is not a silver bullet to data integration costs in smart environments and the cost of integrating with legacy data should not be underestimated. This is of relevance within enterprise settings where the non-technical challenges (e.g. sharing data among departments) can be as significant as the technical ones. See Chap. 2 for further discussion on these challenges.

**The 5 Star Pay-As-You-Go Model Simplified Communication with Non-technical Users** The 5 star pay-as-you-go model for data management (see Chap. 4 for more details) was particularly useful regarding communicating both enhanced functionality and the additional costs of tighter integration with the RLD support services. Within the pilots, it was common to integrate data to the 3 star level on most services. The investment to bring a source to 4 and 5 stars was only made for core datasets within a pilot, and not for each service. Interestingly, many datasets that were initially identified in the early design phases as of high importance (e.g. sensor specifications, detailed infrastructure schematics) remained at the 1 star level as they were not needed by the final applications developed. This resulted in significant savings by avoiding unnecessary integration costs. Within the commercial pilots where more legacy data was available, the 5 star model supported the articulation of the business case for the investments necessary to include data sources and the level of their integration in the dataspace.

**A Secure Canonical Source for Entity Data Simplifies Application Development** Programmable access to the catalog by enabling queries over the machine-readable metadata and entities was crucial to facilitate application development in the dataspace. The role of the catalog and EMS as a canonical source for identifiers for entities was critical to managing the entities in the dataspace. Demonstrating the secure query capability of the access control service was essential to get "buy-in" and build trust with the pilot data owners. For example, the sensor data within the domestic pilot was sensitive, and we needed to assure the residents it was secured so that only privileged users could access their sensor data.

**Data Quality with Things and Sensors Is Challenging in an Operational Environment** Data quality challenges are further complicated as participating data sources, and things within the RLD are not under its full control. Data quality issues included incorrect file formats, incorrect timestamps, unusual sensor usage values, multiple and conflicting values, and missing data. Specifically, concerning the timestamps, the different time zones of pilot sites in different countries posed a challenge, as well as the time changes due to Daylight Saving Time. Keeping raw data where possible, allowed these issues to be addressed and for the analysis to be rerun with the data quality issues resolved. Finally, physical access to the infrastructure can be a significant challenge within operational environments. Within the Linate airport pilot, the infrastructure was often underground within secured parts of the airport. One cannot rely on having physical access to restart or update infrastructure. As a result, the system design must be fault tolerant and adapt to operating conditions.

**Working with Three Pipelines Adds Overhead** The complexity of maintaining the RLDs' three different processing pipelines (the batch, real time, entity layers of the entity-centric query services, see Chap. 10) was challenging concerning the engineering and operational overhead involved. Diagnosing problems and faults required the workflow of all pipelines to be checked for issues, and this can increase the time needed to resolve a problem. A possible future direction is to look at end-to-end exactly-once stream processing technologies (Kappa Architectures). However, the highly decentralised nature of a smart environment and the lack of end-to-end control within dataspaces may not be suitable to the additional coordination/control overhead of exactly-once stream processing approaches. This is an area of future work.

## 16.6 Summary

In this chapter, we reflect on the experience of developing different IoT-based intelligent applications and digital twins within five different smart environments, from Airport to Schools, where the goal has been to engage users within IoT-based intelligent systems to increase water and energy awareness, management, and conservation. The overall design philosophy has been guided using Boyd's "OODA Loop" for decision-making. The chapter detailed the role of a Real-time Linked Dataspace and its support services to enable the creation of intelligent applications and digital twins. The effectiveness of intelligent applications and digital twins within the pilots is evaluated to determine the level of savings achievable. The evaluation identified significant savings within the evaluation period at all the pilot sites. Finally, we reflected on our experiences using the RLD and captured these as a set of lessons learnt.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

# Chapter 17
# A Model for Internet of Things Enhanced User Experience in Smart Environments

**Edward Curry, Willem Fabritius, Souleiman Hasan, Christos Kouroupetroglou, Umair ul Hassan, and Wassim Derguech**

**Keywords** User experience design · Transtheoretical Model · Behaviour change · Internet of Things · Energy management · Water management · Dataspaces

## 17.1 Introduction

Smart environments have emerged in the form of smart cities, smart buildings, smart energy, smart water, and smart mobility [24], where the Internet of Things (IoT)-based infrastructure can support the efficient use of resources within the environment (e.g. water, energy, and waste) [289]. To this end, smart environments can engage a wide range of end users with different interests and priorities, from corporate managers looking to improve the performance of their business to school children who want to explore and learn more about the world around them. Creating a compelling user experience within a smart environment (from smart buildings to smart cities) is an essential factor to success. In this chapter, we reflect on our experience of developing intelligent applications using a Real-time Linked Dataspace within a smart airport, office, home, mixed-use, and school, where the goal has been to engage a wide range of users (from building managers to business travellers) to increase water and energy awareness, management, and conservation.

This chapter explores the use of a Real-time Linked Dataspace in the context of delivering enhanced user experiences. Section 17.2 details a model for delivering an Internet of Things (IoT)-enhanced user experience within a smart environment. The use of the Transtheoretical Model of behaviour change to guide a user's journey to improve their sustainability is explained in Sect. 17.3. Section 17.4 details specific intelligent applications that were developed using the dataspace to support users on their journey guided by the Transtheoretical Model. Section 17.5 details the user study and the results achieved in the pilots. The chapter ends with lessons learnt from our experiences in the pilot deployments in Sect. 17.6 and a summary in Sect. 17.7.

© The Author(s) 2020

E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_17

## 17.2 A Model for Internet of Things Enhanced User Experience

A key challenge in delivering smart environments is creating a compelling user experience with new digital infrastructures within the environment. We assess this challenge within the Physical-Cyber-Social (PCS) computing paradigm [14] that supports a richer human experience with a holistic data-rich view of smart environments that integrate, correlate, interpret, and provide contextually relevant abstractions to humans. Building useful Internet of Things (IoT) applications for smart environments requires the combination of technology, techniques, and skills from multiple disciplines, including electronic engineering, data engineering, and data science, to user experience design and behavioural science.

In Chap. 16, we used the *Observe*, *Orient*, *Decide*, and *Act* (OODA) decision loop to provide a framework to structure the different types of data management support that intelligent applications needed from the dataspace. In this chapter, we take a user-centric perspective of a smart environment that builds on our previous work [16] by formalising the model and providing enhanced details on the intelligent applications created. In Fig. 17.1, we illustrate a model to structure the landscape that

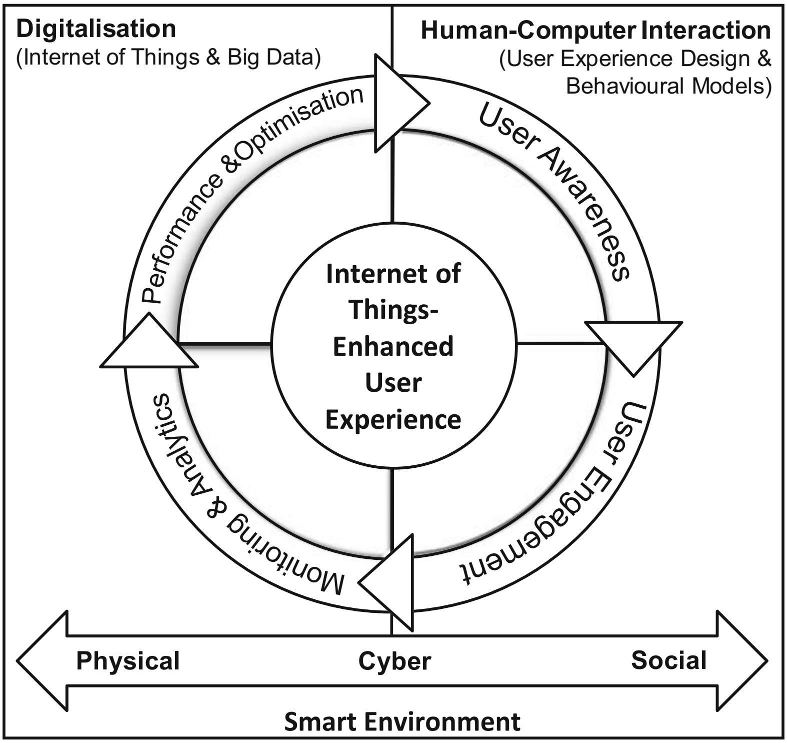

Digitalisation
(Internet of Things & Big Data)
Human-Computer Interaction
(User Experience Design &
Behavioural Models)
Performance &Optimisation
User Awareness
Internet of Things-Enhanced User Experience
Monitoring & Analytics
User Engagement
Physical
Cyber
Social
Smart Environment

**Fig. 17.1** A model for IoT-enhanced user experience

is broken down into two parts, one with a focus on Digitalisation of the Environment using IoT and Big Data, and the other on Human–Computer Interaction (HCI), with a focus on the users' journeys using behavioural models and user experience design [16].

### *17.2.1 Digitalisation: IoT and Big Data*

The digitalisation side of the model is primarily focused on (*Monitoring and Analysis*) using IoT platforms and Big Data processing infrastructure that collects and analyses the data from the smart environment. It is essential to follow a systematic approach to information gathering, analysis, visualisation, and decision-making within a smart environment. This data can then be used within data analytics and decision support (e.g. predictive and perspective analytics, and simulations) that support the *Performance and Optimisation* of the smart environment (e.g. reduced energy or water usage) [31]. These are common steps used in the creation of digital twins, as discussed in Chap. 16.

### *17.2.2 Human–Computer Interaction: IoT-Enhanced User Experience and Behavioural Models*

The Human–Computer Interaction (HCI) side is where we look at how the data and insights generated from the digitalisation side can be used to provide a seamless, personalised IoT-enhanced user experience; providing the right data to the right users at the right time. The key activities on this side of the model look to increase *User Awareness* using targeted information delivery via personalised usage dashboards and task-oriented applications. *User Engagement* with the smart environment is through alerts, notifications, or spatial tasks where users are requested to take physical actions in the environment in the form of citizen actuation [265].

When designing intelligent applications for humans, it is important to consider well-established guidelines and best practices for HCI. For example, within the context of intelligent energy and water management, it is necessary to study the design of conservation interventions in the workplace, user preferences for information visualisations, and the psychology of persuasion and motivation (among others). We will now detail how this model can be used to develop IoT-enhanced user experience for intelligent energy and water systems.

## 17.3 An IoT-Enhanced Journey for Smart Energy and Water

There is a significant opportunity to improve energy and water resource management and conservation by analysing, designing, and implementing intelligent systems to increase demand and supply efficiency. To manage energy and water holistically within a smart environment, it is essential to use decision support tools that present meaningful and contextual information about usage, price, and availability of energy and water intuitively and interactively to users. Smart environments by leveraging IoT can support the development of intelligent applications for efficient and effective management of the resource within the environment. Users will need different information to manage their energy and water consumption, from home users managing their personal water usage, business users managing the water consumption of their commercial activities, to municipalities managing regional distribution and consumption at the level of a city or a region. To deliver a completing user experience for these diverse users, it is necessary to leverage knowledge from several different sources, from IoT-based devices to contextual data sources (including environmental impacts, water quality, energy usage, end-user feedback, occupancy patterns, and meteorological data). This mass of data needs to be analysed to extract insights which then must be packaged within a set of personalised applications, designed within the context of a holistic user journey for the targeted user.

Over the past years, we have been involved in a number of projects [18, 62, 63] concerned with investigating the use of next-generation information platforms for smart environments, specifically targeting intelligent energy and water management systems. The five pilot smart environments are detailed in Chap. 14. Within these projects, we have developed user applications following the IoT-enhanced model of user experiences detailed in the previous section (Sect. 17.2). Within our pilots, the model was implemented using a Real-time Linked Dataspace for the digital-side and behaviour change theories for the HCI-side (see Fig. 17.2).

### *17.3.1 Digital: Real-time Linked Dataspace*

To manage a resource sustainably, it is essential to follow a systematic approach to information gathering, analysis, visualisation, and decision-making. Critical to successful resource management is the visualisation of that data, and its use in decision-making to reduce consumption. Intelligent systems and decision support tools need to present meaningful and contextual information about usage, price, and availability in an intuitive and interactive manner. This requires a data platform that is capable of bringing together multiple dynamic and contextual data sources from the smart environment [1] (see Chap. 2). In this book, we advocate the use of the dataspace paradigm in the design of data platforms to support data ecosystems for intelligent systems.

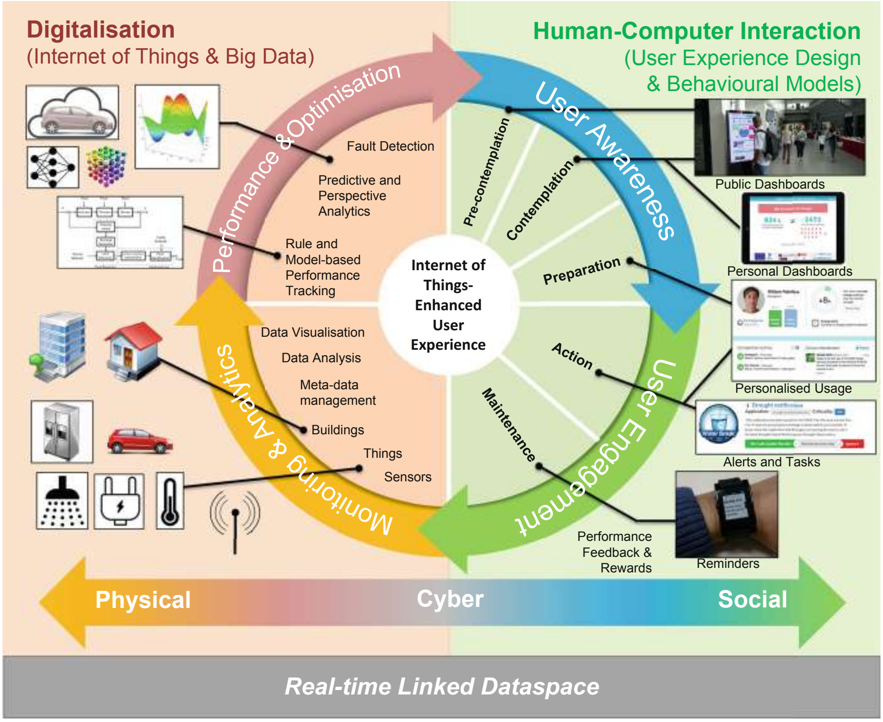


**Fig. 17.2** A model for IoT-enhanced user experience. Adapted from [16]

A dataspace is a data management approach that recognises that in large-scale integration scenarios, involving thousands of data sources, it is difficult and expensive to obtain an upfront unifying schema across all sources [2]. We have created the Real-time Linked Dataspace (RLD) as a data platform for intelligent systems within smart environments. The RLD combines the pay-as-you-go paradigm of dataspaces with linked data, knowledge graphs and real-time stream and event processing capabilities to support a large-scale distributed heterogeneous collection of streams, events, and data sources [4]. The RLD has support services specifically designed to support the management and processing of data from IoT-based smart environments and further details on the RLD is available in Chap. 4.

### *17.3.2 HCI: A User's Journey to Sustainability Using the Transtheoretical Model*

A key aspect of reducing water and energy usage is increasing user awareness about of their resource usage and changing their consumption behaviour. At the core of the HCI-side of the model, we leverage behaviour change theories. The central assumption behind attitudinal theories of behaviour change is that by influencing a person's attitude positively towards a behaviour, they will subsequently act it out.

The 40-year history of Environmental Psychology research has provided a wealth of theoretical models and best practices for influencing sustainable behaviour. What remains a substantial challenge for designers in the HCI community, however, is the translation of these theories into useful and engaging experiences that have the potential to influence behaviour in a meaningful and long-lasting way. Many eco-feedback designs researched within the HCI community have lacked a theoretical connection to established psychological theory (from a recent review, it was less than half of the papers surveyed [342]).

#### 17.3.2.1 Transtheoretical Model

As a framework with which to bridge multiple strands of behaviour change theory, the Transtheoretical Model (TTM) can be used as a guiding heuristic for the high-level design of the user experience. Developed by Prochaska et al. [343, 344], the TTM describes the "stages of change" a person goes through when modifying their behaviour. The model has been developed and applied primarily within the field of healthcare, for example, in exercise and addiction treatment. The TTM has also been researched as a framework for energy feedback technology design [345]. Below is a list of the TTM stages:

- *Pre-contemplation ("Not Ready")*: User is unaware that their behaviour is problematic.
- *Contemplation ("Getting Ready")*: User is aware of the problem and the desired behaviour change with an understanding of the pros and cons of their continued actions.
- *Preparation ("Ready")*: User intends to take action in the immediate future and may begin taking small steps towards behaviour change.
- *Action ("Doing")*: User is undertaking the desired behaviour.
- *Maintenance ("Check")*: User works to sustain the desired behaviour change.

We use the TTM within the HCI side of our model to help identify user informational needs and appropriate persuasion strategies at each stage of change, acting as a guiding design heuristic. It should be noted that other models of behaviour change could also be considered, including the Behaviour Change Model (BCM) for sustainability by Geller [346]. BCM can be described as more focused on user behaviours and needs and is specific to the sustainability context. The TTM was preferred due to its perspective from the user's personal experience, which was useful in considering the user journey. Some applications focused on social influence and gamification strategies, and the TTM was seen as being more flexible when guiding design decisions outside of those aspects focused on sustainability.

#### 17.3.2.2 User Journey

Using the TTM, we defined the user journey scenarios, as illustrated in Fig. 17.3. This strategy helps to consider the activities of users at multiple stages of engagement with intelligent energy and water systems. The user's journey map is:

- *User is unaware of the problem (Pre-contemplation)*: Create awareness about the issue. Highlight social norms, and the benefits of changing behaviour. Ensure a balanced argument and limited detail.

  *Example intervention*:

– Receive an email invitation:
  "As an office worker, I want to receive information about how I can participate in the new energy and water saving initiative within my workplace."

- *User is aware of the problem and the desired behaviour change (Contemplation)*: Make a case for using the system. Appeal to values, and use persuasion strategies such as loss aversion, cognitive dissonance, and foot in the door technique.

  *Example intervention*:

– Visit the promotional page:
  "As an office worker, I want to learn what the system does and how it can benefit my organisation and me."

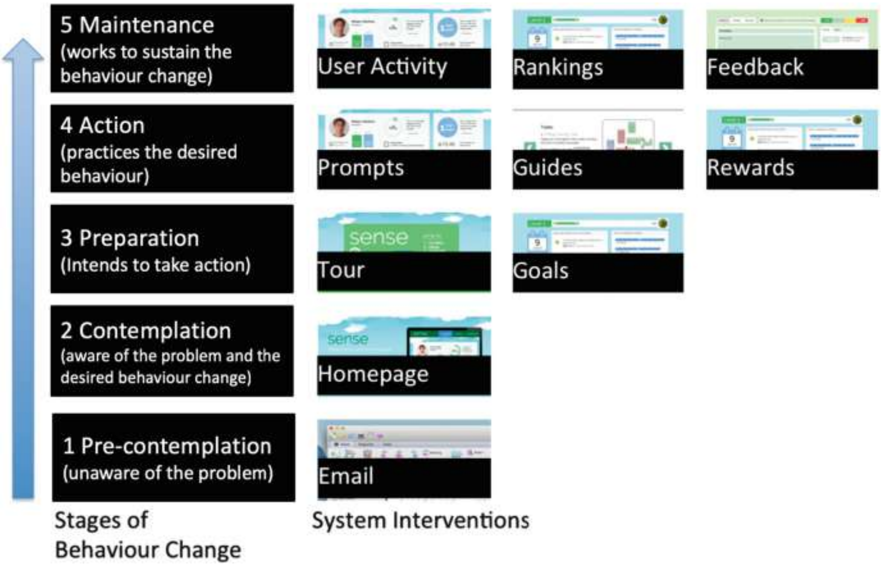


**Fig. 17.3** Stages of behaviour change in the Transtheoretical Model aligned with interventions

- *User intends to take action (Preparation)*: Help users plan for change. Implement persuasion strategies such as goal setting and commitment. Provide support through mentoring.

  *Example intervention:*

- First-time access to the dashboard:
  "As an office worker, I want to learn how to use the website and start saving energy within my workplace."

- *User practices the desired behaviour (Action)*: Provide timely feedback and positive reinforcement for targeted actions. Encourage intrinsic motivation through personalisation.

  *Example interventions*:

- Automatically assigning tasks to optimise energy usage:
  "As an office worker, I want to perform tasks assigned by the system so that I can help save energy in my building."
- Automatically assigning tasks to monitor the environment:
  "As an office worker, I want to help the system in monitoring my building environment so that it can make better decisions about energy saving."
- Automatically assign tasks to maintain occupant comfort:
  "As an office worker, I want to use the system to monitor my building temperature, to help maintain a comfortable working environment."

- *User works to sustain the behaviour change (Maintenance)*: Help users form new habits. Use reminders and feedback towards goals. Encourage mentoring of others and keep journals.

  *Example interventions*:

- View energy consumption feedback:
  "As an office worker, I want to use the system to monitor the energy consumption of my room and see the effect of my energy saving actions."
- View competition feedback:
  "As an office worker, I want to use the system to monitor my personal and team's progress within the energy saving competition."
- Suggest a new energy saving goal:
  "As an office worker, I want to use the system to share new ways of saving energy with my co-workers."

- *Relapse*: Relapse between stages can happen at any time.

## 17.4 TTM Intelligent Applications

Across the five pilot sites, we developed 25 different intelligent applications to support users to optimise resource usage from highly technical leakage detection applications for building managers, to the personal dashboard for office workers and children at home and at school. The process started with design examples of conservation systems within the HCI literature [342] and the commercial sector. User experience tests helped improve the final designs and revealed a high level of engagement from the users. To illustrate this process, we present the intelligent applications developed for the Smart Office Pilot in Galway.

In this pilot, the intelligent system was called SENSE, and it focused on the management of energy within the smart office, mainly targeting the office workers of the building. The following user interfaces were designed for SENSE users to support all the use case intervention scenarios described in the previous section, which follow the stages outlined in the TTM [343]. The three primary SENSE user interfaces we will detail are:

- *Promotional Homepage*: An information resource for describing the SENSE system.
- *Dashboard Tour*: A tour page providing a walk-through of the SENSE dashboard.
- *SENSE Dashboard*: A web application consisting of four sections, including Energy overview, Personal status summary, Community participation, and Tasks.

Each of these user interfaces supports the user journey identified in Sect. 17.3, by providing personalised and relevant information to the user. Table 17.1 details the five stages of the TTM aligned with the user journey together with suggested interventions [345] and the applications we developed for each stage.

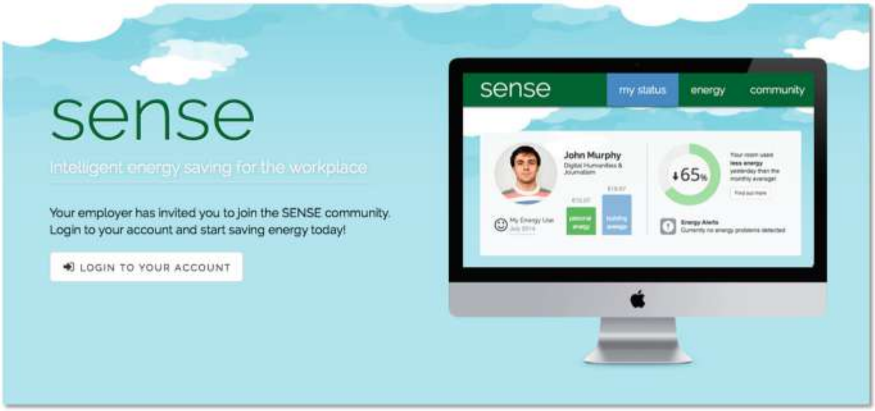
### *17.4.1 Promotional Homepage*

The promotional homepage serves as an informational resource, helping new users to learn about the SENSE system (see Fig. 17.4). Through its design and the information provided, it seeks to establish credibility with the user and gain their trust, helping to bring users from the contemplation stage to the preparation stage of the TTM-based user journey. Design techniques used to establish credibility follow the guidelines outlined by Fogg [347]. The homepage aims to answer common questions new users may have such as: What is SENSE about? What can I do with SENSE? How does it work? What should I do next?

The homepage is limited to providing a general overview of the SENSE system when a user would like to learn more about how to engage with the site; they are encouraged to login and take a tour of the SENSE dashboard. For returning users, the homepage serves as a login portal, with easy access to the login dialogue provided.

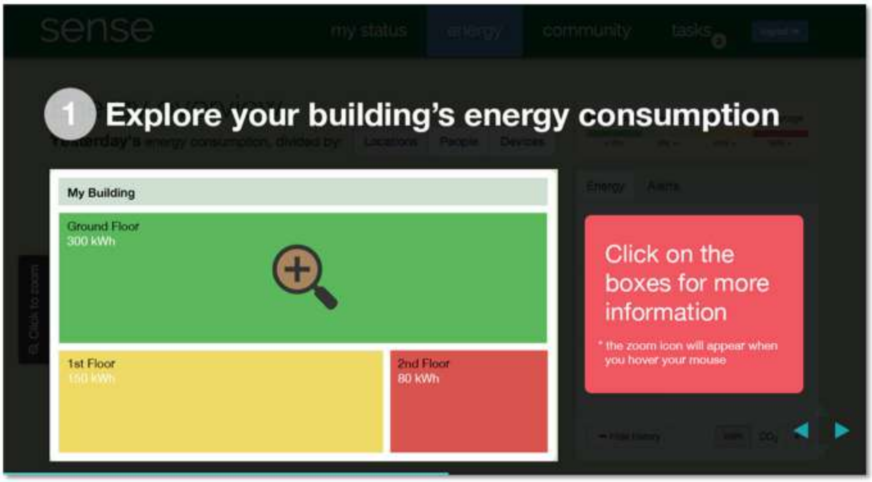
**Table 17.1** Stages of behaviour change in the Transtheoretical Model aligned with interventions. The five stages of the Transtheoretical Model are used to define a user journey for smart energy and water environments. Adapted from [16]

| TTM stage | User journey | User-centric intervention [345] | SENSE applications |
|---|---|---|---|
| Pre-contemplation | User is unaware of the problem | Create awareness about the issue. Highlight social norms, and the benefits of changing behaviour. Ensure a balanced argument and limited detail. | –Public dashboard<br>–Newsletters |
| Contemplation | User is aware of the problem and the desired behaviour change | Make a case for using the system. Appeal to values, use persuasion strategies such as loss aversion, cognitive dissonance, and foot in the door technique. | –Personalised dashboard |
| Preparation | User intends to take action | Help users plan for change. Implement persuasion strategies such as goal setting and commitment. Provide support through mentoring. | –Personalised dashboards<br>–Tour<br>–Goals |
| Action | User practices the desired behaviour | Provide timely feedback and positive reinforcement for targeted actions. Encourage intrinsic motivation through personalisation. | –Personalised dashboards<br>–Alerts Tasks<br>–Guides,<br>–Rewards |
| Maintenance | User works to sustain the behaviour change | Help users form new habits. Use reminders and feedback towards goals. Encourage mentoring of others, keep journals. Relapse between stages can happen at any time. | –User activity<br>–Rankings<br>–Performance feedback and reminders |



Your employer has invited you to join the SENSE community. Login to your account and start saving energy today!

**Fig. 17.4** The homepage top panel with a purpose statement and login button



1 Explore your building's energy consumption

Click on the boxes for more information

* the zoom icon will appear when you hover your mouse

**Fig. 17.5** Tour describing the interaction with the building energy overview

### *17.4.2 Dashboard Tour*

The tour section aims to provide users with a brief overview of the SENSE dashboard. It supports the user in the preparation stage of the TTM-based user journey by providing a walk-through of each section, intending to help users to understand the building energy consumption and personal status feedback provided (see Fig. 17.5). The content of the tour includes:

- Understanding the meaning of personal energy consumption feedback in the *My Status* section.
- Interpreting and interacting with the building overview visualisation in the *Energy* section.
- The game mechanics underlying the *Energy Saving* competition
- Using task guides for performing energy saving (citizen actuation) and room inspection tasks.

The tour supports multiple modes of interaction to progress through the presentation, including mouse clicking the slide or using the keyboard arrow keys, space bar, or enter key. These interaction channels ensure that users do not spend significant time learning new interactions for a likely one-time activity. To allow for the use of the entire screen when presenting information, the visible controls are minimal arrows confined to the bottom-right corner of the screen.

### *17.4.3 Sense Dashboard*

The SENSE Dashboard plans an important role for multiple stages of the user journey from contemplation and preparation to action and maintenance. The dashboard consists of four sections, including *Energy Overview, Personal Status Summary, Community Participation,* and *Tasks.*

#### 17.4.3.1 Energy Overview

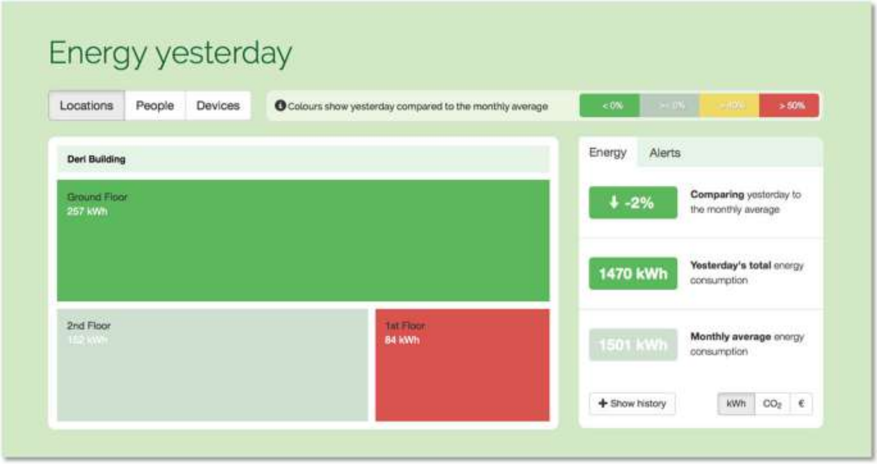
The energy overview section grew out of user requirements for detailed energy consumption reporting. Requirements outlined by users for displaying the energy data included:

- Showing a breakdown of energy consumption by building occupants, location, and devices.
- The ability to monitor energy consumption over specific time intervals (e.g. 1 day, 1 week).
- Showing trends and highlighting abnormal energy consumption to building occupants.
- Displaying energy consumption metrics in Kilowatt Hours, $CO_{2,}$ and Euros.

These requirements are primarily following the user research regarding preferences for energy information visualisations in the workplace [348, 349]. The energy overview section provides three perspectives on energy use within a building, each seeking to answer specific questions, examples of which are:

- *People-centric (*e.g. *personal activity, teams)*:
  - How am I doing compared to others?
  - When is my energy consumption the highest?
  - Which is the best performing team in the building?
- *Location-centric (*e.g. *desk, room, floor, building)*:
  - Which rooms are the most significant consumers/problem areas?
  - How does my room compare to other rooms?
- *Device-centric (*e.g. *lighting, heating, sockets)*:
  - What device is consuming the most energy in my room? When does this occur?
  - Which devices are causing energy waste problems in my building?

To this end, several visualisations were researched to answer the questions outlined above. The primary challenge for selecting a visualisation was its ability to display a wide variety of information in a condensed area while still maintaining ease of learning. After consideration, the zoomable Treemap visualisation design



Colours show yesterday compared to the monthly average

**Comparing** yesterday to the monthly average

**Yesterday's total** energy consumption

**Monthly average** energy consumption

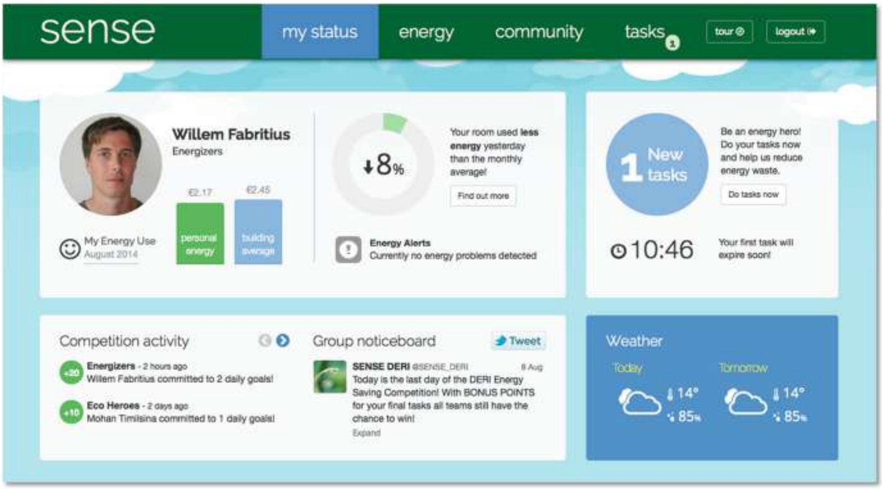
**Fig. 17.6** Energy consumption Treemap visualisation

(see Fig. 17.6) was chosen. The Treemap design, originally developed by Ben Schneiderman [350], follows the Visual Information Seeking Mantra, described as “providing an overview first, support zoom and filter interactions and provide details on demand” [351]. The design has gained broader appeal in recent years and has been recommended as a promising solution for displaying energy consumption trends [349].

### 17.4.3.2 Personal Status Summary

The personal status summary page provides a personalised overview of the user’s relationship with energy and their engagement in the SENSE system, including personalised feedback about the user’s energy consumption, the status of their room, the current tasks assigned to them and the activity occurring within their work community (see Fig. 17.7). The personal status summary is central to the action and maintenance stages of the TTM-based user journey.

The panel in Fig. 17.8a compares the user’s personal energy consumption to that of their colleagues, utilising the Social Proof strategy outlined by Cialdini [352]. The financial cost of the user’s personal energy consumption is contrasted with that of the average person in the building. Costs are aggregated over a calendar month, allowing users to track their progress while also having a normative influence (Social Proof [352]). Visual feedback is alternated depending on the status of this comparison. For example, if the user’s consumption is higher, their personal energy bar is coloured red and the face displayed beside their name is given a neutral expression. When consumption is below average, the colour is green, and a smiling expression is used. These indicators invoke Social Norms which are both descriptive (showing monetary values) and injunctive (showing moral judgement through facial expressions)



Your room used **less energy** yesterday than the monthly average!

**Energy Alerts**
Currently no energy problems detected

Be an energy hero! Do your tasks now and help us reduce energy waste.

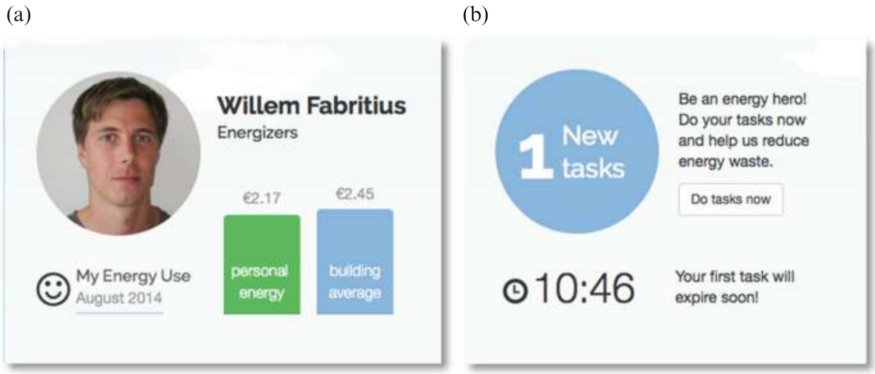
Your first task will expire soon!

**Energizers** - 2 hours ago
Willem Fabritius committed to 2 daily goals!

**Eco Heroes** - 2 days ago
Mohan Timilsina committed to 1 daily goals!

**SENSE DERI** @SENSE_DERI 8 Aug
Today is the last day of the DERI Energy Saving Competition! With BONUS POINTS for your final tasks all teams still have the chance to win!

**Fig. 17.7** Personal status summary



Be an energy hero! Do your tasks now and help us reduce energy waste.

Your first task will expire soon!

**Fig. 17.8** (**a**) Personal energy consumption panel (**b**) Task notification panel

[352]. It is important to note that negative facial expression is not used due to observations made by [353] when using this feedback strategy, where complaints were received from customers.

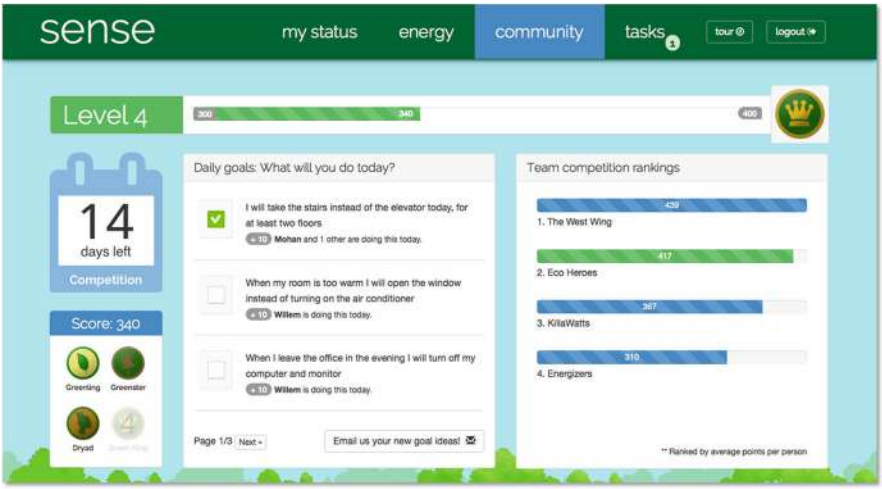
The prominence and proximity of the user's name and photo aim to highlight that the energy consumption recorded is a personal reflection of themselves within their organisation. This can trigger cognitive dissonance [354] when a user's personal image of being environmentally conscious does not match with the message portrayed. Also, the prominence of the team name appeals to the user's sense of relatedness, a strategy for establishing Intrinsic Motivation [355].

The task notification panel in Fig. 17.8b is the most important section of the Personal Status summary. This panel serves as a call to action for users to perform tasks, and it employs a variety of techniques to attract their attention. Firstly, when a task is available, the panel uses a flashing animation to attract attention. This type of visualisation creates a strong response [356] and should be used sparingly; therefore, after 5 s, the animation ceases. Secondly, the expiry time is highlighted to the user with a countdown clock animation. This indicates the limited window of opportunity for the user to perform the task, employing the strategy of Scarcity [352]. Finally, the use of "labelling" feedback appeals to the user's sense of identity ("Be an Eco-Hero") [357] eliciting desirable associations of eco-friendliness (Biospheric values [358]). This message could equally be adapted to elicit associations of being a team player (Altruistic) or money saver (Egoistic), depending on the user's personal values [358].

### 17.4.3.3 Community Participation

The community participation section (see Fig. 17.9) is important for the maintenance stage of the TTM-based user journey as it provides a platform on which to run energy saving competitions among teams in the workplace. The interface uses a "points, badges and leaderboard" system (commonly known as the PBL gamification strategy [359]). By appealing to the user's sense of competence, autonomy, and relatedness, intrinsic motivation can be increased, encouraging users to participate in an activity for their own inherent pleasure (e.g. the fun factor) [355]. For example, by encouraging users to form teams and engage in friendly competition with their co-workers, a sense of relatedness is created [355].

Extrinsic motivators are also involved, such as a user's desire for rewards and status [360]. We will now explore three key panels in this section for daily goals, team competitions, and rewards.



☑ I will take the stairs instead of the elevator today, for at least two floors
+10 Mohan and 1 other are doing this today.

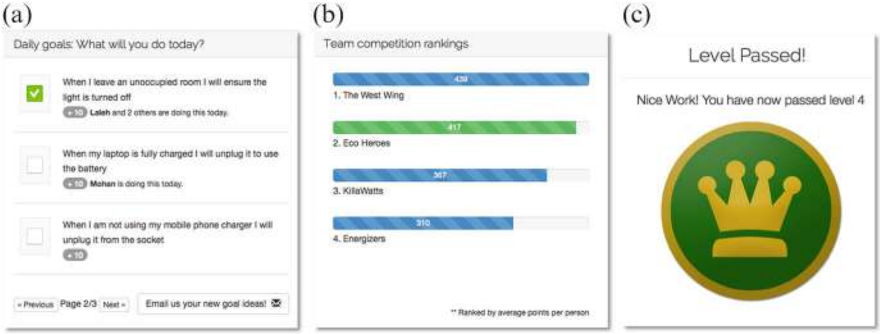
☐ When my room is too warm I will open the window instead of turning on the air conditioner
+10 Willem is doing this today.

☐ When I leave the office in the evening I will turn off my computer and monitor
+10 Willem is doing this today.

1. The West Wing
2. Eco Heroes
3. KillaWatts
4. Energizers

** Ranked by average points per person

Fig. 17.9 Community participation section



Daily goals: What will you do today?

☑ When I leave an unoccupied room I will ensure the light is turned off
+10 **Laleh** and 2 others are doing this today.

□ When my laptop is fully charged I will unplug it to use the battery
+10 **Mohan** is doing this today.

□ When I am not using my mobile phone charger I will unplug it from the socket
+10

« Previous Page 2/3 Next » Email us your new goal ideas! ✉

Team competition rankings

430
1. The West Wing
417
2. Eco Heroes
367
3. KillaWatts
310
4. Energizers

** Ranked by average points per person

Level Passed!

Nice Work! You have now passed level 4

**Fig. 17.10** (**a**) Daily goals panel, (**b**) Team competition rankings panel, (**c**) Rewards panel

**Daily Goals Panel** This panel [see Fig. 17.10a] allows users to publicly commit to energy saving goals, which is shown to be an effective motivation strategy [361–365]. The goals listed address common issues of energy waste in office environments; activities that are generally easy to perform and accepted as “best practices” [366]. By committing to a goal, users earn points which help to improve their personal score and team rankings. Users are required to visit the website daily to make these commitments, helping to drive traffic to the website and serving as a reminder. Participants can submit new goal suggestions via the email link at the bottom-right of the panel. Allowing users to select and suggest their own goals appeals to the users’ sense of autonomy, an essential factor in intrinsic motivations [355]. The goals’ written format utilises the Implementation Intentions strategy [367], whereby anticipated situational cues are used as an anchor to trigger the desired behaviour [368] (e.g. “When I leave the office I will turn off my computer and monitor”).

The names of other users who have committed to a goal are displayed within the goals panel, encouraging influence through the social proof strategy [352]. Also, the knowledge that the users’ own commitments will be made public, increases the likelihood of compliance with the stated goals, by appealing to their desire to show the consistency of character within their peer group [352].

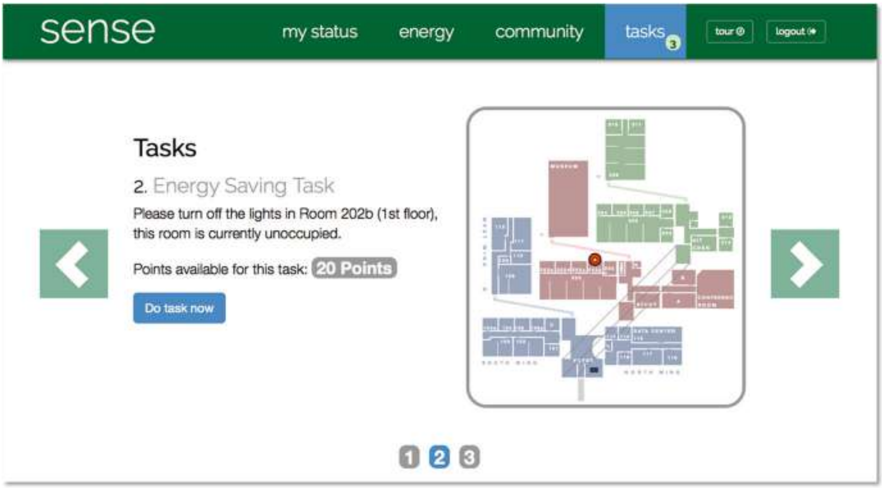
**Team Competition Rankings Panel** The team rankings panel [see Fig. 17.10b] allows users to receive feedback on their team’s status within the energy saving competition, appealing to a user’s competitive nature and desire for status [360]. By focusing on team level rankings, users are not singled out by their peers in terms of performance, which was a significant concern highlighted in the user research conducted by Foster et al. [369]. Comparative feedback among teams has been shown to be effective in previous research on workplace energy consumption reduction [370]. Displaying a set time limit for the competition using a calendar utilises the Scarcity principle [352] for user motivation.

**Rewards Panel** Competitions are shown to appeal more strongly to certain personality types as well as personal achievements to others (e.g. using Bartle's Character Theory this maps to Killers and Achievers, respectively [371]). Providing rewards such as points and badges [see Fig. 17.10c] can appeal to these types, with a sense of achievement coming from level progression as opposed to outperforming other players [360]. The level progress bar [see Fig. 17.10c] allows users to receive immediate feedback on their progress, appealing to a user's sense of competence [355] and providing positive re-enforcement [372] (Fogg's Conditioning principle [373]).

#### 17.4.3.4 Tasks

Two types of tasks are available in the smart office pilot: (1) data management tasks and (2) citizen actuation tasks for energy savings. These tasks are implemented using the human task service of the RLD. Further details on this service are available in Chap. 9, and an example of an energy saving task is provided later in this chapter and also in Chap. 16.

The tasks interface is arguably the most important section in the dashboard, helping to guide user actions to solve energy waste problems directly as part of the action stage of the user journey. The interface utilises Fogg's Suggestion and Tunnelling principles [373], by prompting users and providing task instructions to guide them through the task. The task instructions include a building map for identifying the location of the task (Fig. 17.11), a task instruction gallery with a step-by-step walk-through of how to perform a task (Fig. 17.12) and form inputs describing the data required for collection (Fig. 17.13).



Tasks

2. Energy Saving Task

Please turn off the lights in Room 202b (1st floor), this room is currently unoccupied.

Points available for this task: 20 Points

Do task now

**Fig. 17.11** Energy saving task description

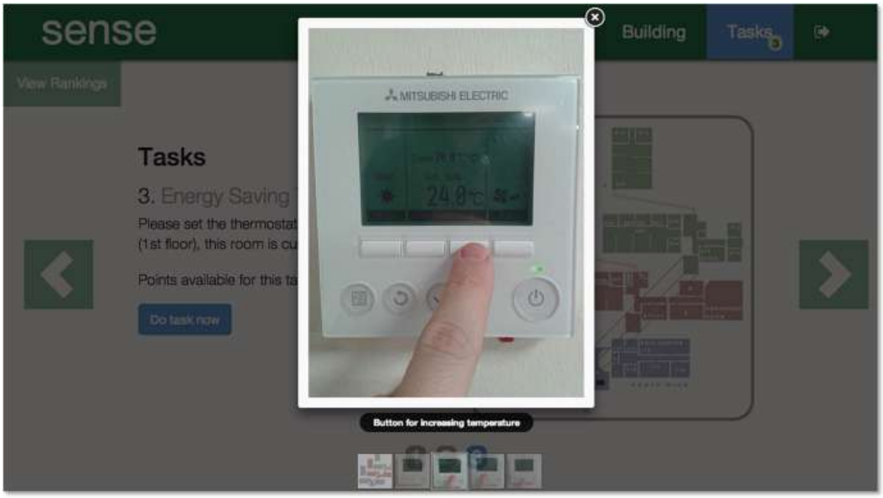
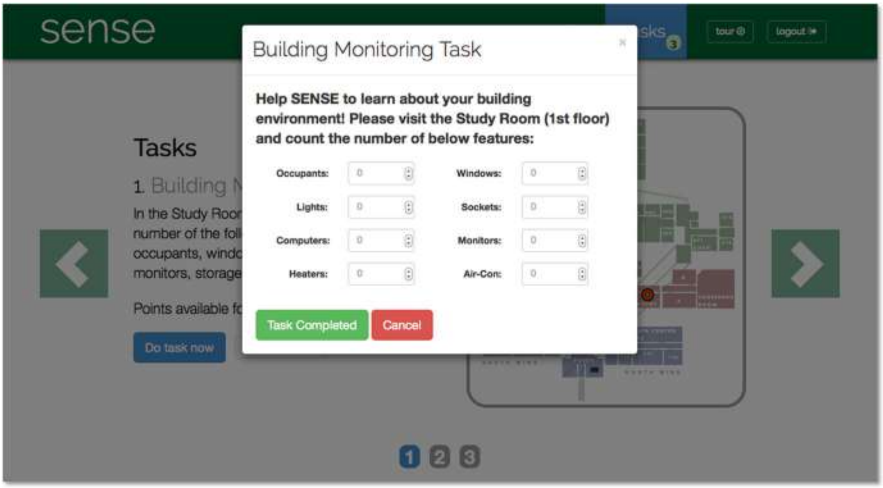

sense
Building
Tasks
3
View Rankings
Tasks
3. Energy Saving
Please set the thermostat
(1st floor), this room is cu
Points available for this ta
Do task now
MITSUBISHI ELECTRIC
24.0°C
Button for increasing temperature

Fig. 17.12 Occupant comfort task instruction gallery



Building Monitoring Task

Help SENSE to learn about your building environment! Please visit the Study Room (1st floor) and count the number of below features:

| | | | |
|---|---|---|---|
| Occupants: | 0 | Windows: | 0 |
| Lights: | 0 | Sockets: | 0 |
| Computers: | 0 | Monitors: | 0 |
| Heaters: | 0 | Air-Con: | 0 |

Task Completed Cancel

Fig. 17.13 Room inspection task form

## 17.5 User Study

This section presents the results from deploying the model for IoT-enhanced user experience in the smart environments described in Chap. 14. In this section, we detail the methodology used within each pilot, the energy and water savings achieved in the pilots, and the changes in user awareness at the pilots.

### 17.5.1 Methodology

In general, the pilots followed a similar methodology for the design, deployment, and evaluation of the system. The high-level research methodology followed during the evaluation of the deployed intelligent systems is summarised in Table 17.2. During the initial period of the pilots, metering data was collected from existing systems to establish baselines across all pilots. During the control period, the users within the pilots had access to the data generated by the metering infrastructure system through traditional information systems (e.g. Building Management System, and basic public dashboards within the airports, office building, and school). The data collection period for each pilot spanned between 6 and 16 months, which also included a range of user interventions such as pre-surveys, focus groups, interviews,

**Table 17.2** Research methodology for pilot sites. Adapted from [16]

| Pilot Site | Location | Users | Study period | Baseline | Method |
|---|---|---|---|---|---|
| Smart Airport | Linate Airport, Italy | Corporate users<br>Passengers | 10 months | Pre-study water and energy usage | Passenger survey<br>User trials<br>Staff questionnaires |
| Smart Office | Insight, Ireland | 150 Office workers | 6 months | Average energy usage | User study (11 participants)<br>Field study (6 participants)<br>User trials (4–6 participants) |
| Smart Homes | Thermi, Greece | Domestic users<br>Utility providers | 16 months | Manually recorded monthly water usage | Preliminary questionnaires (8 participants)<br>Focus groups<br>Interviews |
| Mixed Use | National University of Ireland Galway, Ireland | University students<br>Staff & management<br>Public | 16 months | Pre-study water and energy usage | Pre-intervention survey (110 participants)<br>User trials and feedback cycles<br>Post-intervention survey (110 participants) |
| Smart School | Coláiste na Coiribe, Galway, Ireland | School students<br>Staff & management<br>Public | 12 months | Pre-study water and energy usage | Awareness questionnaire (150 participants)<br>User trials and feedback cycles<br>Post-intervention survey (70 participants) |

and feedback cycles. The user experience evaluation included usability testing, usability field study, and user trials.

### 17.5.2 Impact

The RLD supported the development of more than 25 intelligent applications following the model for IoT-enhanced user experience to serve diverse user groups in five smart environments and provided relevant data for effective data analytics to raise awareness and detect faults. The energy and water saving opportunities identified at the different pilot sites and their estimations in terms of costs and $CO_2$ emissions (see Table 17.3) were significant and convinced building managers and users to take actions and to further expand the approach to other areas (e.g. expand the smart airport pilot to Malpensa Airport).

In terms of increasing user awareness, the goal of the pilot applications was to provide personalised and actionable information about energy and water consumption and availability to individual users intuitively and effectively at a time scale relevant for decision-making. Access to this information helped increase end-user awareness and improved energy and water consumption. As detailed in Table 17.3, the level of user awareness was increased at four out of five pilot sites. In the mixed-use site where there was no increase in awareness, the pre-intervention surveys indicate that the sample populations exhibited a moderately high level of awareness. Post-intervention awareness surveys were found to be statistically insignificant due to the high baseline level of awareness. Since pre-intervention surveys already indicated a high level of awareness, small differences that might or might not have occurred would have been difficult to capture.

**Table 17.3** Impacts on energy and water saving and user awareness for pilot sites. Adapted from [16]

| Pilot Site | Actual savings measured | Estimated annual savings | User awareness |
|---|---|---|---|
| Smart Airport | 2954 $m^3$<br>3013 kg $CO_2$ | 54,000 $m^3$<br>55,080 kg $CO_2$ | Increases awareness of the problem<br>Increase of responsibility and Personal norms |
| Smart Office | 23.86% energy use reduction | – | Increased awareness of usage |
| Smart Homes | 30% water use reduction | – | Increased awareness of usage |
| Mixed Use | 174 $m^3$<br>177 kg $CO_2$ | 8089 $m^3$<br>8251 kg $CO_2$ | Limited increase (high existing awareness baseline) |
| Smart School | 2179 $m^3$<br>2223 kg $CO_2$ | 9306 $m^3$<br>9492 kg $CO_2$ | Increased awareness in teachers<br>Increased awareness of junior students<br>Limited increase in senior students (high existing awareness baseline) |

## 17.6 Insights and Experience Gained

Based on a reflection of our experience across the pilot sites, the following lessons were identified as key learnings on developing IoT-enhanced user experiences. They can be used to inform the design of the user experience of future intelligent applications within smart environments.

**Minimise Cognitive Overload with Clear and Focused Applications and Visualisations** Within the pilots, it was shown that participants preferred applications and visualisations that had a low cognitive load. Complex applications with full functionality were demanding for users to learn and understand. Users wanted simple (often single purpose) applications over more elaborate multi-purpose ones. We recommended that visualisations and applications be tested early and often to ensure they are easily understood (matching the target users' mental model), and serving the information needs and goals of the target users [374, 375]. Strategies that can be employed to achieve more natural cognition include providing a simplified core message with the ability for users to dig deeper on demand (Progressive Disclosure [374]), harnessing the user's prior experience with design conventions (Consistency [374]) and conceptual knowledge (Priming [374]).

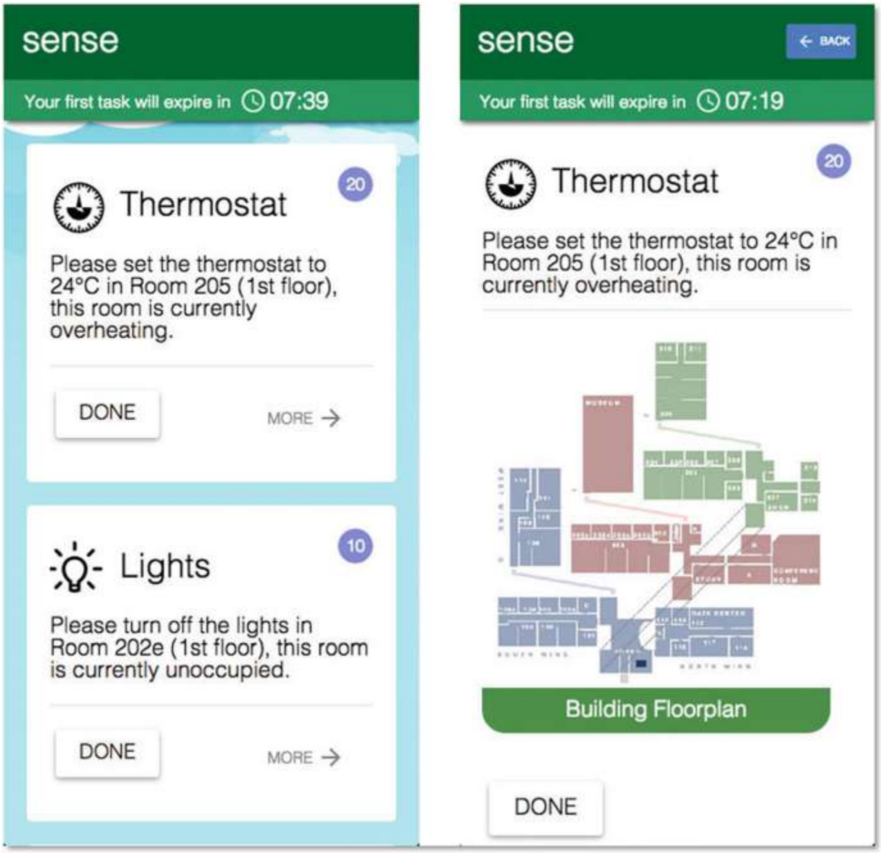
**Understand Your Users' Needs and Their Journey** When designing intelligent applications, consider users and their stage on the journey. Customising the applications to support a specific task or action helped to capture their interest and increase engagement. Within our pilots, we delivered 100 "personalised" versions of the 25 different intelligent applications to meet the specific needs of users. For example, building managers operate daily in the "Action" phase of the TTM and are interested in applications with concise messages that help them take immediate actions. Technicians were more interested in task-oriented applications with detailed consumption charts for a dedicated analysis and identification of potential issues in the system. At the other extreme, airport passengers, parents, and kids at the "(pre-) contemplation" phase with a more casual interest wanted applications to help them explore the smart environment to learn more about it.

**Social Influence and Interaction are Strong Motivators** Social influence was shown to be a strong motivator, which is consistent with observations found in the environmental psychology literature, particularly in the workplace [376]. The use of gamification with leaderboards and social benchmarking was effective in the pilots with users enjoying the friendly rivalry and social interaction with peers. However, the critical question remains if this strategy can maintain user interest in the long term? One answer may lie in the theory of intrinsic motivation, in which behaviours are performed for their inherent pleasure and are, therefore, more durable [374].

**Close the Feedback Loop with Personalisation** Within the pilots, users had a strong desire to have responsive feedback regarding the impact of their energy and water saving actions, allowing them to track their progress in a closed feedback loop [374]. This strategy appeals to the user's desire for control or mastery, increasing their intrinsic motivation [355]. This feedback was shown to be of most interest when presented as “people-centric”, reporting on the participant’s personal performance (including a comparison to others) and the performance of teams in the competition.

**Bring Your “Humans-in-the-Loop” of the Smart Environment** Users can engage in IoT environments from various perspectives: (1) The user-as-a-consumer where a water or energy management environment collects data on the user consumption and communicates it with them to trigger a behaviour change. (2) The user-as-a-sensor [377], where users can enhance the IoT environment with an image or report on noise or temperature. And (3) the user-as-an-actuator [220], where the user is asked by an IoT platform to take action in the physical world, such as closing a window, based on data collected on temperature drop in winter.

A direct consequence of the human-in-the-loop approach is a sense of ownership, among users, towards the system, leading towards more accurate information and better collaborative management [378]. Figure 17.14 shows an example of an



Your first task will expire in 07:39

Thermostat

Please set the thermostat to 24°C in Room 205 (1st floor), this room is currently overheating.

Lights

Please turn off the lights in Room 202e (1st floor), this room is currently unoccupied.

Your first task will expire in 07:19

Thermostat

Please set the thermostat to 24°C in Room 205 (1st floor), this room is currently overheating.

**Fig. 17.14** Example of an actuation task for collaborative energy management. Chrome application task list page (left) and task detail page (right)

actuation task that requires a user to change the state of the building for energy saving. First, the tasks are generated based on the observed state of the building (e.g. an empty room with lights turned on, or a window open in a room with AC on) through motion and light sensors. Then, it is routed to an occupant in the building who is within the vicinity of the room. The task is also contextualised to provide the required information for performing the task, as well as the associated rewards.

Based on our experience, tasks with short deadlines need to be pushed to users through real-time communication channels (e.g. instant messages); otherwise, tasks can be presented through an appropriate search and browse interface. While some tasks might get users' attention due to their altruistic nature, proper incentives need to be designed and implemented for sustainable participation of users over time (e.g. leaderboards, badges, rewards).

**Careful Use of Targeted Alerts and Notifications** Information overload in IoT-based systems can overwhelm users. To minimise the search friction between actionable information and users, a well-designed notifications system is needed. Emerging technologies and practices in user experience show that notifications will probably play a much more significant role in IoT-based systems [379]. Within the pilots, we aimed to enable notification to attract users' attention only when necessary—furthermore, the notifications needed to deliver actionable information to users. For example, building managers wanted alerts on faults and optimisation opportunities. They had little interest in exploring charts of usage data, often commenting they do not have time to analyse data to gain any insights. In the pilot evaluations, it was found that the frequency of fault alert notifications must be carefully considered. Overwhelming recipients with notifications of potential faults were found to be counter-productive, leading to a potential disregarding of alert messages.

## 17.7 Summary

Creating a compelling user experience within a smart environment (from smart buildings to smart cities) is an essential factor to success. In this chapter, we introduce a model for Internet of Things (IoT)-enhanced user experience for smart environments. The model is broken down into two parts, one with a focus on *Digitalisation* of the Environment using IoT and Big Data, and the other on *Human–Computer Interaction* (HCI), with a focus on the users' journeys using behavioural models and user experience design. We use the model to develop a journey for users within the context of intelligent energy and water systems using the Transtheoretical Model as a guiding heuristic for the high-level user experience design. Several applications were built following the Transtheoretical Model of behaviour change to guide a user's journey to improve their sustainability. The chapter detailed our experience of developing IoT-based intelligent applications within a smart home, school, office building, university, and airport, where the goal was to engage a wide range of users (from building managers to business travellers) to increase water and energy awareness, management, and conservation. The Real-time Linked Dataspace simplified the process of developing these personalised applications by supporting the gathering of data from different sources in the smart environment.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

# Part V
# Future Directions

The final part of this book considers what is required for the widespread adoption of the dataspaces approach within data ecosystems and details a future research agenda for dataspaces, data ecosystems, and intelligent systems.

# Chapter 18
# Future Research Directions for Dataspaces, Data Ecosystems, and Intelligent Systems

**Keywords** Dataspaces · Data ecosystems · Intelligent systems · Research challenges · Technology adoption · Trusted data sharing · Governance · Incremental systems engineering · Human-centricity

## 18.1 Introduction

As we move toward 2030, today's computing paradigms such as data-intensive computing (Big Data), Open Data [380], Knowledge Graphs, Machine Learning, Large-Scale Distributed Systems [381], Internet of Things (IoT), Physical-Cyber-Social Computing [14], Service-Oriented [382], and Cloud/Edge Computing [383] will be the foundations to the realisation of the vision of intelligent systems. In fact, real-world intelligent systems are being enabled by a combination of these paradigms using a mixture of architectures (centralised, decentralised, and a combination of both) and infrastructures such as Middleware and IoT platforms to support the development of intelligent systems and applications [13, 67, 295, 384].

This chapter begins in Sect. 18.2 with an examination of what is required for the widespread adoption of the dataspace approach. Next, in Sect. 18.3 the chapter explores the research landscape towards 2030 by identifying the principal research directions for the dataspaces, data ecosystems, and intelligent systems, including large-scale decentralised support services, multimedia/knowledge-intensive event processing, trusted data sharing, data governance, and economic models, evolving intelligent systems engineering and cognitive adaptability, and finally the path towards human-centric systems. The chapter finishes with a summary in Sect. 18.4.

© The Author(s) 2020

E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0_18

## 18.2 Dataspaces: From Proof-of-Concept to Widespread Adoption

Our vision for the intelligent systems of 2030 is that they will be a significant part of a dynamic global data ecosystem where vast amounts of data can move among actors within complex information supply chains [1, 25]. Users, applications, and machines will still need to leverage these data flows to optimise physical and virtual systems in the areas of economy, environment, energy, water, waste, people (intellectual endowment and engagement), built environment, mobility (transportation), and public spaces [385, 386]. We believe dataspaces will be a crucial technology platform, enabling users to harness the data from the global data ecosystem. In order to deliver the potential of dataspaces, the wide-scale diffusion of the technology is necessary. The literature on the diffusion of innovations and technology adoption [387] can assist in understanding how this could happen.

In their study of technology change, Anderson and Tushman [388] argue that technology progresses in a series of cycles, hinging on technological discontinuity followed by a design competition which results in the emergence of a dominant design. According to Anderson and Tushman, the dominant design is never in the same form as the first discontinuity, and it is not on the leading edge of technology; it bundles features to meet the requirements of most of the market. Once a dominant design emerges, organisations often cease to invest in learning alternative designs and instead focus on developing competencies related to the dominant design.

Understanding these cycles and patterns can indicate as to the trends in the data management domain and the potential to improve the adoption of the dataspace paradigm. The first wave of dataspace initiatives [2, 87, 179] can be seen as a large-scale design competition consisting of Proof-of-Concept projects that explored the potential for dataspaces within specific data management use cases. The goal was to understand the requirements, explore the design space, and discover the boundaries of the many different support services needed to enable the dataspace data management paradigm. The early dataspaces were designed and developed by world-leading researchers and required high levels of expertise. The defining characteristics of many projects in this wave was a focus on the experimental design together with a pilot deployment (e.g. biomedical, energy [100], personal [87, 88, 91, 92]) to meet specific data management requirements.

The second wave of dataspace initiatives, now underway [4, 101], is focusing more on general deployments of dataspaces beyond the specific initial use cases to drive broader adoption. The key challenge in this wave is the need to identify the dominant design needed to support the requirements of mass-market adoption. The innovation adoption literature can again guide dataspace researchers in improving the uptake of their technology. The likelihood of an innovation being adopted can be increased if it possesses specific key characteristics [389]. The following criteria have been adapted from [389] for the context of a dataspace within a data ecosystem:

- *Relative Advantage:* Enabling a better functioning data ecosystem and usage of data within the ecosystem.
- *Compatibility:* The degree to which a dataspace is consistent with existing stakeholder values, or interests, and usage context.
- *Complexity:* The degree of difficulty involved in implementing the dataspace and communicating benefits to stakeholders.
- *Trialability:* The degree to which experimentation is possible with a dataspace.
- *Cost Efficiency and Feasibility:* Concerning existing comparable data management practices.
- *Evidence:* The availability of research evidence and practical efficacy of a dataspace.
- *Risk:* Level of risk associated with the implementation and adoption of a dataspace.

Dataspaces will need to possess a number of these characteristics if they are to be successfully adopted within the general data management domain. This sets out a clear research direction for next-generation dataspaces.

## 18.3 Research Directions

Dataspaces are a relatively new research area that brings together several other areas in computer science and other disciplines. We now discuss research areas [1, 4] which are essential to enabling the next-generation of dataspaces, data ecosystems, and intelligent systems (see Fig. 18.1). Research is needed to overcome many challenges, including decentralised support services, support for multimedia data, trusted data sharing, governance and economic models, incremental systems engineering, and human-centricity.

### *18.3.1 Large-Scale Decentralised Support Services*

As dataspaces are deployed at larger scales, it will be necessary to create enhanced support services, to scale entity management, and to minimise the cost of operation for these deployments [4]. Challenges include:

- *Enhanced Supported Services:* Many enhancements are possible for support services of dataspaces including the use of natural language interfaces to improve user experience [390], decentralised support services for large-scale deployments [391], and privacy-by-design [392] approaches to support the ever-increasing amount of personal information captured in intelligent systems.
- *Scaling Entity Management:* Within larger-scale deployments, it will be necessary to enhance the entity management services to support both the increase in entities, data, and users. Ranking and summarisation need to be query- and

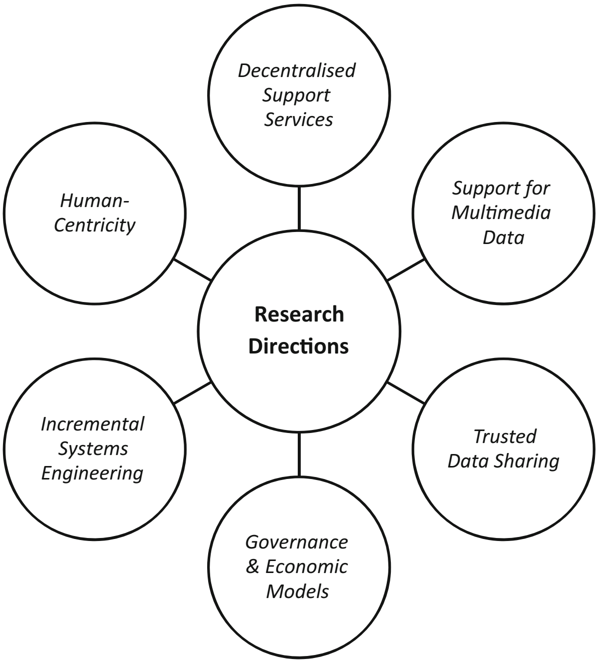

Decentralised Support Services
Human-Centricity
Support for Multimedia Data
Research Directions
Incremental Systems Engineering
Trusted Data Sharing
Governance & Economic Models

**Fig. 18.1** Research directions for dataspaces, data ecosystems, and intelligent systems

activity-relevant [86], with relevant facts, but at the same time diverse to cater for a wide range of information/conceptualisation need. A trade-off between processing time and expressiveness is necessary. Furthermore, there is significant potential for extensive usage of large-scale crowdsourcing for "human-in-the-loop" data management and curation [71, 117].

- *Maintenance and Operation Cost:* As the size of deployment increases, it will be necessary to investigate new techniques to improve the performance of dataspaces in terms of the maintenance and operational costs of the support platform within large-scale deployments (e.g. city-level) [26].

## 18.3.2 Multimedia/Knowledge-Intensive Event Processing

As multimedia streams become more pervasive with the Internet of Multimedia Things (IoMT), it will be necessary for dataspaces to provide specific support services to process and manage these streams. New techniques and approaches will be needed for:

- *Support Services for Multimedia Data:* As multimedia data becomes more common in data ecosystems through the IoMT, there will be a direct need for appropriate support services within dataspaces. There is an opportunity to leverage advances in deep learning for image processing (e.g. object detection) [30], which can be the basis of dataspace support services for rich content types including text and multimedia streams.

- *Placement of Multimedia Data and Workloads:* The increased computing resources needed to process, and extract multimedia data will pose challenges for existing techniques for processing and data placement. This will require dataspace support services to consider the simultaneous training and processing of multimedia streams, taking into consideration the geospatial and temporal characteristics of smart environments.
- *Adaptive Training of Classifiers:* To effectively process multimedia data, dataspaces will need to be able to assemble the appropriate classifiers to extract features from multimedia content based on the needs of users of the dataspace at runtime. The training of classifiers needs to be adaptable to the changing requirements of data ecosystems. There is a need to support transfer learning among intelligent systems and for collective efforts to build pre-trained models for datasets and to bootstrap dataspace support services. Finally, distributed approaches to training classifiers are needed to maximise the available resources from the cloud to the edge of the network.
- *Complex Multimedia Event Processing:* To detect patterns from multimedia streams within a dataspace, it will be necessary to investigate new techniques for complex multimedia event processing. Challenges include defining the language to express the complexity of the event patterns and the content of the event, optimisation techniques to improve system performance for event detection over computationally intensive multimedia streams, and methods to train models over incoming media streams for new unseen queries in lack of available training data.

### *18.3.3 Trusted Data Sharing*

There is a need to enable the trusted sharing of data among organisations, people, and systems. This will pose significant challenges for:

- *Trusted Platforms:* A trusted data platform focuses on the secure data sharing among a group of participants (e.g. industrial consortiums sharing private or commercially sensitive data) within a clear legal framework. An ecosystem data platform would have to be infrastructure agnostic and must support continuous, coordinated data flows, seamlessly moving data among systems [1]. Data exchange could be based on models for monetisation or reciprocity. Data platforms can create possibilities for smaller organisations and even individual developers to get access to large volumes of data, enabling them to explore their potential. Trusted platforms open many research areas for dataspaces, including data discovery, curation, linking, synchronisation, standardisation, and decentralisation [25].

- *Usage Control:* The challenges with data sharing go beyond technical issues to issues of data ownership, privacy, business models, smart contracts, and licensing and authorised reuse by third parties. The control paradigm for shared data must shift from today's *access* control to *usage* control, and dataspaces will need to support both of these usage control for both organisations and individuals.
- *Personal Dataspaces:* There is a need for personal dataspaces for the management of the data of the individual. Personal dataspaces will need to respect the relevant legislation for personal data (e.g. General Data Protection Regulation) and allow an individual to remain in control of their personal data and its use. Personal dataspaces will need to balance the need for privacy with the benefits of analytics and handle this trade-off based on the preferences of the individual. Techniques for preserving privacy for metadata, query privacy, and privacy-preserving integration of independent data sources will all be needed in next-generation dataspaces.
- *Industrial Dataspaces:* The sharing of data among commercial organisations will also increase. Industrial dataspaces [101] will be needed to facilitate the trusted and secure sharing and trading of commercial data among collaborating organisations. These platforms will need to provide support services that enable a data marketplace that facilitates the automated licensing of data exchanged among organisations and the enforcement of legal rights and appropriation of remuneration to the original data owners.

### 18.3.4 Ecosystem Governance and Economic Models

For mass collaboration to take place within data ecosystems, we need to overcome the challenges of dealing with large-scale agreements among potentially decoupled interacting parties [1]. New approaches will be needed for:

- *Decentralised Data Governance:* Research is needed on decentralised data governance models for data ecosystems that support collaboration and fully consider ethical, legal, and privacy concerns. Data governance within a data ecosystem must recognise data ownership, sovereignty, and regulation while supporting economic models for the sustainability of the data ecosystem [1]. A range of decentralised governance approaches may guide a data ecosystem from authoritarian to democratic, including majority voting, reputation models (e.g. eBay), proxy-voting, and dynamic governance (e.g. sociocracy: circles and double linking) [393]. Dataspaces will need to enforce these data governance models automatically.
- *Economic Models:* Economic model may be used as an incentivisation factor within governance models including support for "data-vote exchange" models (pay for votes with data), and economic models for peer-to-peer systems [394, 395]. The sharing and exchange of data within dataspaces could also be based on models for monetisation or reciprocity. Data platforms can create

possibilities for smaller organisations and even individual developers to get access to large volumes of data, enabling them to explore their potential [1].

### *18.3.5 Incremental Intelligent Systems Engineering: Cognitive Adaptability*

The design of adaptive intelligent systems will need to consider the implication of operating within a smart environment and its associated data ecosystem [1]. This will pose significant challenges for systems engineering:

- *Pay-As-You-Go Systems:* The boundaries of systems will be fluid and will change and evolve at run-time to adapt to the context of the current situation. However, we must consider the cost of system participation, and support “pay-as-you-go” approaches at both the system and data levels [1]. How can the pay-as-you-go approach of dataspaces be extended to the design of incremental and evolving systems? How can we integrate systems on an “as-needed” basis with the labour-intensive aspects of system integration postponed until they are required?
- *Cognitive Adaptability:* Work on evolving systems engineering [29] will need to consider the inclusion of data-driven probabilistic techniques that can provide “Cognitive Adaptability” to help intelligent systems adapt to changes in the environment that were unknown at design-time by enriching the control-loop with observational data from the environment. Intelligent system designers will need to consider the varying levels of accuracy offered by data-driven approaches, providing best-effort or approximate results using the data accessible at the time [1]. How can we mix deterministic and statistical approaches in the design of intelligent systems? How can we test and verify these systems? There is a need to support transfer learning among intelligent systems and for joint efforts to build pre-trained models for system adaptability. Dataspaces can play a role in supporting these collective efforts.

### *18.3.6 Towards Human-Centric Systems*

Currently, intelligent systems make critical decisions in highly engineered systems (e.g. autopilots) where users receive specialised training to interact with them (e.g. pilots). As we move forward, intelligent systems will be making both critical and lifestyle decisions: from the course of treatment for a critical illness, safely driving a car, to choosing what takeout to order and the temperature of our shower [1]. This will pose specific challenges in the design of human-centric systems:

- *Explainable Artificial Intelligence and Data Provenance:* Data-driven decision approaches (including Cognitive and AI-based techniques) will need to provide explanations and evidence to support their decisions and guarantees for the decisions they recommend. How can we trust the large-scale, data-driven decision-making provided by dataspace-powered AI platforms? This will require a greater need for provenance support within dataspaces to support the audit trail necessary to justify a data-driven decision [396].

- *Human-in-the-Loop:* The role of users in intelligent systems will not be a passive one. Users are a critical part of socio-technical systems, and we need to consider more ways of including the "Human in the Loop" within future intelligent systems. Active participation of users can improve their engagement and sense of ownership of the system. Indeed, active involvement of the user could be a condition for them granting access and usage of their private data. Research is needed to give trust in algorithms and data, in the trusted co-evolution between humans and AI-based systems, and in the legal, ethical, and privacy issues associated with making data-driven critical decisions [1].

## 18.4 Summary

This chapter examined what is required for the widespread adoption of the dataspace approach. The chapter then explored the future research landscape by identifying the principal research directions for the dataspaces, data ecosystems, and intelligent systems including large-scale decentralised support services, multimedia/knowledge-intensive event processing, trusted data sharing, data governance and economic models, evolving systems engineering and cognitive adaptability, and finally the path towards human-centric systems.

**Open Access** This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

# Credits

I would like to extend my sincere gratitude and thanks to the following authors and publishers for permission to include material in the text:

- Reprinted from *Future Generation Computer Systems*, vol. 90, Curry, E., Derguech, W., Hasan, S., Kouroupetroglou, C., and ul Hassan, *A Real-time Linked Dataspace for the Internet of Things: Enabling 'Pay-As-You-Go' Data Management in Smart Environments*, pp. 405–422, Copyright 2019, with permission from Elsevier (Materials found in Chaps. 1, 2, 4, 6, 9, 10, 14, 16, and 18).
- © 2018 IEEE. Reprinted, with permission, from Curry, E., and Sheth, A. 2018. "Next-Generation Smart Environments: From System of Systems to Data Ecosystems," *IEEE Intelligent Systems* (33:3), pp. 69–76 (Materials found in Chaps. 2 and 18).
- With permission of the Authors. M. Franklin, A. Halevy, and D. Maier, "From databases to dataspaces: a new abstraction for information management," *ACM SIGMOD Rec.*, vol. 34, no. 4, pp. 27–33, 2005. (Materials found in Chap. 3).
- Reprinted/adapted by permission from Springer, "Dataspaces" in *Search Computing*, by Hedeler, C., Belhajjame, K., Paton, N. W., Campi, A., Fernandes, A. A. A., and Embury, S. M. 2010 (Materials found in Chap. 3).
- © 2012 IEEE. Reprinted, with permission, from Freitas, A., Curry, E., Oliveira, J. G., and O'Riain, S. 2012. "Querying Heterogeneous Datasets on the Linked Data Web: Challenges, Approaches, and Trends," *IEEE Internet Computing* (16:1), pp. 24–33 (Materials found in Chap. 7).
- Derguech, W., Bhiri, S., Hasan, S., and Curry, E., Using Formal Concept Analysis for Organizing and Discovering Sensor Capabilities, *The Computer Journal, 2015,* 58, 3, pp. 356–367, by permission of Oxford University Press (Materials found in Chap. 8).
- Reprinted from *Expert Systems with Applications,* vol. 58, ul Hassan, U., and Curry, E, Efficient task assignment for spatial crowdsourcing: A combinatorial fractional optimization approach with semi-bandit learning, pp. 36–56, Copyright 2016, with permission from Elsevier (Materials found in Chap. 9).

© The Author(s) 2020
E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0

- Reprinted from *Future Generation Computer Systems,* vol. 76, Gao, F., Ali, M. I., Curry, E., and Mileo, A., Automated discovery and integration of semantic urban data streams: The ACEIS middleware, pp. 561–581, Copyright 2017, with permission from Elsevier (Materials found in Chap. 11).
- © 2015 IEEE. Reprinted, with permission, from Qin, Y., Sheng, Q. Z., and Curry, E. 2015. “Matching Over Linked Data Streams in the Internet of Things,” *IEEE Internet Computing,* 19:3, pp. 21–27 (Materials found in Chap. 12).
- © 2013 IEEE. Reprinted, with permission, from D. Crowley, E. Curry, and J. Breslin, “Closing the Loop – From Citizen Sensing to Citizen Actuation,” in *7th IEEE International Conference on Digital Ecosystems and Technologies*, 2013 (Materials found in Chap. 16).
- © 2018 IEEE. Reprinted, with permission, from Curry, E., Hasan, S., Kouroupetroglou, C., Fabritius, W., ul Hassan, U., and Derguech, W. 2018. “Internet of Things Enhanced User Experience for Smart Water and Energy Management,” *IEEE Internet Computing* (22:1), pp. 18–28 (Materials found in Chap. 17).

# References

1. E. Curry, A. Sheth, Next-generation smart environments: from system of systems to data ecosystems. IEEE Intell. Syst. **33**(3), 69–76 (2018)
2. M. Franklin, A. Halevy, D. Maier, From databases to dataspaces: a new abstraction for information management. ACM SIGMOD Rec. **34**(4), 27–33 (2005)
3. M. Stonebraker, U. Çetintemel, S. Zdonik, The 8 requirements of real-time stream processing. ACM SIGMOD Rec. **34**(4), 42–47 (2005)
4. E. Curry, W. Derguech, S. Hasan, C. Kouroupetroglou, U. ul Hassan, A real-time linked dataspace for the internet of things: enabling 'pay-as-you-go' data management in smart environments. Futur. Gener. Comput. Syst. **90**, 405–422 (2019)
5. K.B. Mankad, An intelligent process development using fusion of genetic algorithm with fuzzy logic, in *Nature-Inspired Computing* (IGI Global, 2017), pp. 312–348
6. G. Hulten, *Building Intelligent Systems* (2018)
7. U. Köse, A. Arslan, Chaotic systems and their recent implementations on improving intelligent systems, in *Handbook of Research on Novel Soft Computing Intelligent Algorithms: Theory and Practical Applications* (2014), pp. 69–101
8. M. Weiser, R. Gold, J.S. Brown, The origins of ubiquitous computing research at PARC in the late 1980s. IBM Syst. J. **38**(4), 693–696 (1999)
9. L. Roffia et al., Requirements on system design to increase understanding and visibility of cultural heritage, in *Handbook of Research on Technologies and Cultural Heritage: Applications and Environments* (2011)
10. A. Astaras, H. Lewy, C. James, A. Katasonov, D. Ruschin, P.D. Bamidis, Unobtrusive smart environments for independent living and the role of mixed methods in elderly healthcare delivery: the USEFIL approach, in *Handbook of Research on Innovations in the Diagnosis and Treatment of Dementia* (2015)
11. G.M. Youngblood, E.O. Heierman, L.B. Holder, D.J. Cook, Automation intelligence for the smart environment, in *IJCAI International Joint Conference on Artificial Intelligence* (2005)
12. D.J. Cook, S.K. Das, *Smart Environments: Technology, Protocols and Applications* (2005)
13. L. Baresi, L. Mottola, S. Dustdar, Building software for the internet of things. IEEE Internet Comput. **19**(2), 6–8 (2015)
14. A. Sheth, P. Anantharam, C. Henson, Physical-cyber-social computing: an early 21st century approach. IEEE Intell. Syst. **28**(1), 78–82 (2013)
15. E. Curry, The big data value chain: definitions, concepts, and theoretical approaches, in *New Horizons for a Data-Driven Economy: A Roadmap for Usage and Exploitation of Big Data in Europe*, ed. by J.M. Cavanillas, E. Curry, W. Wahlster (Springer International Publishing, 2016)

© The Author(s) 2020

E. Curry, *Real-time Linked Dataspaces*,
https://doi.org/10.1007/978-3-030-29665-0

16. E. Curry, S. Hasan, C. Kouroupetroglou, W. Fabritius, U. ul Hassan, W. Derguech, Internet of things enhanced user experience for smart water and energy management. IEEE Internet Comput. **22**(1), 18–28 (2018)
17. L. Atzori, A. Iera, G. Morabito, The internet of things: a survey. Comput. Networks **54**(15), 2787–2805 (2010)
18. C.C. Aggarwal, N. Ashish, A. Sheth, The internet of things: a survey from the data-centric perspective, in *Managing and Mining Sensor Data* (2014), pp. 383–428
19. A.Y. Halevy, M.J. Franklin, D. Maier, Dataspaces: a new abstraction for information management, in *International Conference on Database Systems for Advanced Applications*. LNCS vol. 3882 (2006), pp. 1–2
20. D. Laney, 3D data management: controlling data volume, velocity, and variety. Appl. Deliv. Strateg., 2 (2001)
21. M. Loukides, What is data science? *O'Reily Radar* (June 2010)
22. A. Jacobs, The pathologies of big data. Commun. ACM **52**(8), 36 (2009)
23. C.W. Choo, The knowing organization: how organizations use information to construct meaning, create knowledge and make decisions. Int. J. Inf. Manage. **16**(5), 329–340 (1996)
24. E. Ahmed, I. Yaqoob, A. Gani, M. Imran, M. Guizani, Internet-of-things-based smart environments: state of the art, taxonomy, and open research challenges. IEEE Wirel. Commun. **23** (5), 10–16 (2016)
25. J.M. Cavanillas, E. Curry, W. Wahlster, *New Horizons for a Data-Driven Economy: A Roadmap for Usage and Exploitation of Big Data in Europe* (Springer International Publishing, Cham, 2016)
26. E. Curry, S. Dustdar, Q.Z. Sheng, A. Sheth, Smart cities: enabling services and applications. J. Internet Serv. Appl. **7**(1), 6 (2016)
27. ISO/IEC/IEEE, *ISO/IEC/IEEE 15288: 2015 Systems and Software Engineering – System Life Cycle Processes* (2015)
28. M.W. Maier, Architecting principles for systems-of-systems. Syst. Eng. **1**(4), 267–284 (1998)
29. M. Hinchey, L. Coyle, Evolving critical systems: a research agenda for computer-based systems, in *2010 17th IEEE International Conference and Workshops on Engineering of Computer Based Systems* (2010), pp. 430–435
30. A. Aslam, E. Curry, Towards a generalized approach for deep neural network based event processing for the internet of multimedia things. IEEE Access **6**, 25573–25587 (2018)
31. S. Rusitschka, E. Curry, Big data in the energy and transport sectors, in *New Horizons for a Data-Driven Economy: A Roadmap for Usage and Exploitation of Big Data in Europe*, ed. by J.M. Cavanillas, E. Curry, W. Wahlster (Springer International Publishing, 2016)
32. B. Marr, What is digital twin technology – and why is it so important? *Forbes* (2017)
33. B. Cheng, S. Longo, F. Cirillo, M. Bauer, E. Kovacs, Building a big data platform for smart cities: experience and lessons from Santander, in *Proceedings – 2015 IEEE International Congress on Big Data, BigData Congress 2015* (2015)
34. O. Vermesan et al., Internet of things strategic research and innovation agenda, in *Internet of Thing Converging Technologies for Smart Environment and Integrated Ecosystems*, (2013)
35. M. Cilia, C. Bornhövd, A.P. Buchmann, Moving active functionality from centralized to open distributed heterogeneous environments, in *9th International Conference on Cooperative Information Systems (CooplS '01)* (2001)
36. L. Von Bertalanffy, *General System Theory* (1968)
37. P.R. Carlile, Transferring, translating, and transforming: an integrative framework for managing knowledge across boundaries. Organ. Sci. **15**(5), 555–568 (2004)
38. C.E. Shannon, W. Weaver, *A Mathematical Theory of Communication* (University of Illinois Press, Urbana, IL, 1949)
39. A. Freitas, *Schema-Agnostic Queries for Large-Schema Databases: A Distributional Semantics Approach* (National University of Ireland, Galway, 2015)
40. C. Bizer, T. Heath, T. Berners-Lee, Linked data - the story so far. Int. J. Semant. Web Inf. Syst. **5**(3), 1–22 (2009)

41. T. Berners-Lee, Linked data-design issues (2006), https://www.w3.org/DesignIssues/LinkedData.html
42. M. Arenas, A. Bertails, E. Prud'hommeaux, J.F. Sequeda, A direct mapping of relational data to RDF. W3C Recommendation, 27 September 2012
43. T. Heath, C. Bizer, Linked data: evolving the web into a global data space. Synth. Lect. Semant. Web Theory Technol. **1**(1), 1–136 (2011)
44. H. Paulheim, Knowledge graph refinement: a survey of approaches and evaluation methods. Semant. Web **8**(3), 489–508 (2016)
45. M. Nickel, K. Murphy, V. Tresp, E. Gabrilovich, A review of relational machine learning for knowledge graphs. Proc. IEEE **104**(1), 11–33 (2016)
46. M. Treacy, F. Wiersema, Customer intimacy and other value disciplines. Harv. Bus. Rev. **71** (1), 84–93 (1993)
47. J. Weinman, *Digital Disciplines: Attaining Market Leadership via the Cloud, Big Data, Social, Mobile, and the Internet of Things* (2015)
48. F.A. Zeleti, A. Ojo, E. Curry, Exploring the economic value of open government data. Gov. Inf. Q. (2016)
49. B. Metcalfe, Metcalfe's Law after 40 years of ethernet. Computer (Long. Beach. Calif.) **46** (12), 26–31 (2013)
50. A.G. Tansley, The use and abuse of vegetational concepts and terms. Ecology **16**(3), 284–307 (1935)
51. J.F. Moore, Business ecosystems and the view from the firm. Antitrust Bull. **51**, 31–75 (2006)
52. J.F. Moore, Predators and prey: a new ecology of competition. Harv. Bus. Rev. **71**, 75–86 (1993)
53. J.F. Moore, *The Death of Competition: Leadership and Strategy in the Age of Business Ecosystems* (HarperCollins, New York, 1996)
54. S. Gossain, G. Kandiah, Reinventing value: the new business ecosystem. Strateg. Leadersh. **26** (5), 28–33 (1998)
55. R. Adner, Match your innovation strategy to your innovation ecosystem. Harv. Bus. Rev. **84** (4), 98–107 (2006)
56. S. O'Riáin, E. Curry, A. Harth, XBRL and open data for global financial ecosystems: a linked data approach. Int. J. Account. Inf. Syst. **13**(2), 141–162 (2012)
57. H. Kim, J.-N. Lee, J. Han, The role of IT in business ecosystems. Commun. ACM **53**(5), 151 (2010)
58. E. Curry, A. Freitas, S. O'Riáin, The role of community-driven data curation for enterprises, in *Linking Enterprise Data*, ed. by D. Wood, (Springer US, Boston, MA, 2010), pp. 25–47
59. G. Koenig, Business ecosystems revisited. Management **15**(2), 208–224 (2012)
60. A. Gawer, M.A. Cusumano, Industry platforms and ecosystem innovation. J. Prod. Innov. Manag. **31**(3), 417–433 (2014)
61. M.D. Wilkinson et al., The FAIR Guiding Principles for scientific data management and stewardship. Sci. Data **3**, 160018 (2016)
62. E. Curry, J. O'Donnell, E. Corry, S. Hasan, M. Keane, S. O'Riain, Linking building data in the cloud: Integrating cross-domain building data using linked data. Adv. Eng. Informatics **27**(2), 206–219 (2013)
63. E. Curry et al., Linked water data for water information management, in *11th International Conference on Hydroinformatics (HIC 2014)* (2014)
64. D. Pfisterer et al., SPITFIRE: toward a semantic web of things. IEEE Commun. Mag. **49**(11), 40–48 (2011)
65. D. Puiu et al., CityPulse: large scale data analytics framework for smart cities. IEEE Access **4**, 1086–1108 (2016)
66. J. Soldatos et al., OpenIoT: open source internet-of-things in the cloud, in *International Workshop on Interoperability and Open-Source Solutions for the Internet of Things* (2015), pp. 13–25

67. J.M. Schleicher, M. Vögler, S. Dustdar, C. Inzinger, Enabling a smart city application ecosystem: requirements and architectural aspects. IEEE Internet Comput. **20**(2), 58–65 (2016)
68. D. Howe et al., The future of biocuration. Nature **455**(7209), 47–50 (2008)
69. M.L. Brodie, J.T. Liu, The power and limits of relational technology in the age of information ecosystems, in *On the Move to Meaningful Internet Systems OTM (Keynote)* (2010)
70. H.V. Jagadish et al., Big data and its technical challenges. Commun. ACM **57**(7), 86–94 (2014)
71. A. Freitas, E. Curry, Big data curation, in *New Horizons for a Data-Driven Economy: A Roadmap for Usage and Exploitation of Big Data in Europe*, ed. by J. M. Cavanillas, E. Curry, W. Wahlster, (Springer International Publishing, Cham, 2016)
72. C.L. Palmer, M.H. Cragin, P.B. Heidorn, L.C. Smith, Data curation for the long tail of science: the case of environmental sciences, in *3rd International Digital Curation Conference* (2007)
73. J. Musser, T. O'Reilly, *Web 2.0 Principles and Best Practices* (2006)
74. S. Auer, S. Hellmann, The web of data: decentralized, collaborative, interlinked and interoperable, in *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC-2012)* (2012)
75. S. Abiteboul et al., The Lowell database research self-assessment. Commun. ACM **48**(5), 111–118 (2005)
76. P. Helland, If you have too much data, then 'good enough' is good enough. Commun. ACM **54**(6), 40 (2011)
77. P. Pandey, D. Pompili, Exploiting the untapped potential of mobile distributed computing via approximation. Pervasive Mob. Comput. **38**, 381–395 (2017)
78. A. Halevy, M. Franklin, D. Maier, Principles of dataspace systems, in *25th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems – PODS '06* (2006), pp. 1–9
79. M. Singh, A framework for data modeling and querying dataspace systems, in *Seventh International Conference on Data Mining and Warehousing (ICDMW)* (2013)
80. C. Hedeler, K. Belhajjame, N.W. Paton, A. Campi, A.A.A. Fernandes, S.M. Embury, Dataspaces, in *Search Computing*, ed. by S. Ceri, M. Brambilla, (Springer, Heidelberg, 2010), pp. 114–134
81. C. Hedeler et al., Pay-as-you-go mapping selection in dataspaces, in *Proceedings of the 2011 International Conference on Management of Data – SIGMOD '11* (2011), pp. 1279–1282
82. Y. Wang, S. Song, L. Chen, A survey on accessing dataspaces. ACM SIGMOD Rec. **45**(2), 33–44 (2016)
83. A. Das Sarma, X. Dong, A.Y. Halevy, Data modeling in dataspace support platforms, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 5600 (2009), pp. 122–138
84. M. Singh, S.K. Jain, A survey on dataspace, in *Communications in Computer and Information Science* (2011), pp. 608–621
85. E. Curry, System of systems information interoperability using a linked dataspace, in *IEEE 7th International Conference on System of Systems Engineering (SOSE 2012)* (2012), pp. 101–106
86. D.K. Sah, X. Wu, M. Lv, H.T. Mirza, Activity-based linkage and ranking methods for personal dataspace. Int. J. Mob. Commun. **16**(3), 266–285 (2018)
87. J. Dittrich, M. Antonio, V. Salles, iDM: a unified and versatile data model for personal dataspace management, in *32nd International Conference on Very Large Data Bases – VLDB '06* (2006), pp. 367–378
88. J. Liu, X. Dong, Answering structured queries on unstructured data, in *9th International Workshop on the Web and Databases* (2006)
89. X. Dong, A. Halevy, A platform for personal information management and integration, in *CIDR 2005* (2005)

90. J.-P. Dittrich, iMeMex: a platform for personal dataspace management, in *Workshops of International ACM SIGIR Conference on Research and Development in Information Retrieval* (2006), pp. 40–43
91. L. Blunschi, J. Dittrich, O.R. Girard, S. Kirakos, K. Marcos, A.V. Salles, A dataspace odyssey: the iMeMex personal dataspace management system," in *Conference on Innovative Data Systems Research (CIDR)* (2007), pp. 114–119
92. Y. Li, X. Meng, Exploring personal corespace for dataspace management, in *SKG 2009 – 5th International Conference on Semantics, Knowledge, and Grid* (2009)
93. L. Dragan, M. Luczak-Roesch, N. Shadbolt, Understanding personal data as a space-learning from dataspaces to create linked personal data, in *2nd Workshop on Services and Applications over Linked APIs and Data, SALAD 2014, Co-located with the 10th Extended Semantic Web Conference, ESWC 2014* (2014), pp. 18–25
94. R. Grossman, E. Creel, M. Mazzucco, R. Williams, A dataspace infrastructure for astronomical data, in *Data Mining for Scientific and Engineering Applications*, (Springer, Boston, MA, 2001), pp. 115–123
95. A. Hasnain et al., Linked biomedical dataspace: lessons learned integrating data for drug discovery, in *The Semantic Web (ISWC 2014)* (2014), pp. 114–130
96. U. Leser, F. Naumann, (Almost) hands-off information integration for the life sciences, in *Conference on Innovative Database Research CIDR* (2005)
97. M.S. Mota, J.C. dos Reis, S. Goutte, A. Santanchè, Multiscaling a graph-based dataspace. J. Inf. Data Manag. – JIDM **7**(3), 233–248 (2016)
98. Y. Li, C. Hu, Process materials scientific data for intelligent service using a dataspace model. Data Sci. J. **15**, 1–17 (2016)
99. C. Hu, Y. Li, X. Cheng, Z. Liu, A Virtual Dataspaces Model for large-scale materials scientific data access. Futur. Gener. Comput. Syst. **54**, 456–468 (2016)
100. E. Curry, S. Hasan, S. O'Riáin, Enterprise energy management using a linked dataspace for energy intelligence, in *The Second IFIP Conference on Sustainable Internet and ICT for Sustainability (SustainIT 2012)* (2012)
101. B. Otto et al., Industrial data space digital sovereignity over data, in *Fraunhofer-Institut White Paper*, (2016)
102. C. Borjigin, Y. Zhang, C. Xing, C. Lan, J. Zhang, Dataspace and its application in digital libraries. Electron. Libr. **31**(6), 688–702 (2013)
103. J. Madhavan et al., Web-scale data integration: you can only afford to pay as you go, in *Conference on Innovative Database Research CIDR* (2007)
104. M.J. Cafarella, A. Halevy, N. Khoussainova, Data integration for the relational web. Proc. VLDB Endow. **2**(1), 1090–1101 (2009)
105. R. Cyganiak, A. Jentzsch, Linked Open Data cloud diagram (2011)
106. P.A.N. Ying, Y. Tang, Pay-as-you-go software artifacts management. Comput. Informatics **32** (4), 827–843 (2013)
107. F.B. Balint, H.L. Truong, On supporting contract-aware IoT dataspace services, in *Proceedings – 5th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, MobileCloud 2017* (2017)
108. M. Zhong, M. Liu, Q. Chen, Modeling heterogeneous data in dataspace, in *2008 IEEE International Conference on Information Reuse and Integration* (2008), pp. 404–409
109. C. Hedeler et al., DSToolkit: an architecture for flexible dataspace management, in *Transactions on Large-Scale Data- and Knowledge-Centered Systems* (2012), pp. 126–157
110. Y. Li, X. Meng, Supporting context-based query in personal DataSpace, in *Proceeding of the 18th ACM conference on Information and knowledge management – CIKM '09* (2009), p. 1437
111. A. Freitas, E. Curry, J.G. Oliveira, S. O'Riain, Querying heterogeneous datasets on the linked data web: challenges, approaches, and trends. IEEE Internet Comput. **16**(1), 24–33 (2012)
112. R. Grossman, M. Mazzucco, DataSpace: a data Web for the exploratory analysis and mining of data. Comput. Sci. Eng. **4**(4), 44–51 (2002)

113. A. Freitas, J.C.P. da Silva, S. O'Riain, E. Curry, Distributional relational networks, in *AAAI 2013 Fall Symposium on Semantics for Big Data* (2013)
114. F. Mandreoli, A framework for user-driven mapping discovery in rich spaces of heterogeneous data, in *International Conference on Ontologies, DataBases, and Applications of Semantics (ODBASE) 2017*. LNCS vol. 10574 (2017), pp. 399–417
115. U. ul Hassan, S. O'Riain, E. Curry, Leveraging matching dependencies for guided user feedback in linked data applications, in *9th International Workshop on Information Integration on the Web (IIWeb2012)* (2012), pp. 1–6
116. D.W. Archer, L.M.L. Delcambre, D. Maier, A framework for fine-grained data integration and curation, with provenance, in a dataspace, in *TAPP'09 First Workshop on on Theory and Practice of Provenance* (2009)
117. N.A. Azuan, S.M. Embury, N.W. Paton, Observing the data scientist: using manual corrections as implicit feedback, in *2nd Workshop on Human-In-the-Loop Data Analytics (HILDA 2017)* (2017)
118. S.R. Jeffery, M.J. Franklin, A.Y. Halevy, Pay-as-you-go user feedback for dataspace systems, in *Proceedings of the 2008 ACM SIGMOD international conference on Management of data – SIGMOD '08* (2008), pp. 847–860
119. K. Belhajjame, N.W. Paton, S.M. Embury, A.A.A. Fernandes, C. Hedeler, Incrementally improving dataspaces based on user feedback. Inf. Syst. **38**(5), 656–687 (2013)
120. R. McCann, A. Kramnik, W. Shen, V. Varadarajan, O. Sobulo, A. Doan, Integrating data from disparate sources: a mass collaboration approach, in *Proceedings – 21st International Conference on Data Engineering, 2005. ICDE 2005* (2005), pp. 487–488
121. A. Freitas, J.G. Oliveira, S. O'Riain, E. Curry, A multidimensional semantic space for data model independent queries over RDF data, in *Fifth IEEE International Conference on Semantic Computing (ICSC 2011)* (2011)
122. Y. Qin, Q.Z. Sheng, N.J.G. Falkner, A. Shemshadi, E. Curry, Towards efficient dissemination of linked data in the internet of things, in *23rd ACM International Conference on Information and Knowledge Management (CIKM 2014)* (2014), pp. 1779–1782
123. F. Gao et al., QoS-aware complex event service composition and optimization using genetic algorithms, in *International Conference on Service Oriented Computing (ICSOC 2014)*, vol. 8831 (2014)
124. K. Broda, K. Clark, R. Miller, A. Russo, SAGE: a logical agent-based environment monitoring and control system, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2009), pp. 112–117
125. A. Demers, J. Gehrke, M. Hong, M. Riedewald, W. White, Towards expressive publish/subscribe systems, in *10th International Conference on Advances in Database Technology, EDBT'06*, (2006), pp. 627–644
126. F. Wang, P. Liu, Temporal management of RFID data, in *31st International Conference on Very Large Data Bases* (2005), pp. 1128–1139
127. M. Ficco, L. Romano, A generic intrusion detection and diagnoser system based on complex event processing, in *2011 First International Conference on Data Compression, Communications and Processing* (2011)
128. N.W. Paton, O. Díaz, Active database systems. ACM Comput. Surv. **31**(1), 63–103 (1999)
129. D. McCarthy, U. Dayal, The architecture of an active database management system, in *Proceedings of the 1989 ACM SIGMOD International Conference on Management of Data – SIGMOD '89* (1989), pp. 215–224
130. B. Shannon, R. Chinnici, *Java$^{TM}$ Platform, Enterprise Edition (Java EE) Specification, v6* (SUN MICROSYSTEMS, 2009)
131. S. Vinoski, CORBA: integrating diverse applications within distributed heterogeneous environments. IEEE Commun. Mag. (1997)
132. B.H. Tay, A.L. Ananda, A survey of remote procedure calls. ACM SIGOPS Oper. Syst. Rev. **24**(3), 68–79 (1990)

133. D. Garlan, D. Notkin, Formalizing design spaces: implicit invocation mechanisms, in *Formal Software Development Methods (VDM 1991)* (1991), pp. 31–44
134. G. Hohpe, B. Woolf, *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions* (2003)
135. E. Curry, Message-oriented middleware, in *Middleware for Communications*, ed. by Q. H. Mahmoud, (Wiley, Chichester, 2004), pp. 1–28
136. B. Babcock, S. Babu, M. Datar, R. Motwani, J. Widom, Models and issues in data stream systems, in *Proceedings of the Twenty-First ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems – PODS '02* (2002)
137. L. Golab, M.T. Özsu, Issues in data stream management. ACM SIGMOD Rec. **32**(2), 5–14 (2003)
138. S. Babu, J. Widom, Continuous queries over data streams. ACM SIGMOD Rec. **30**(3), 109 (2001)
139. G. Cugola, A. Margara, Processing flows of information. ACM Comput. Surv. **44**(3), 1–62 (2012)
140. D. Luckham, *The Power of Events: An Introduction to Complex Event Processing in Distributed Enterprise Systems* (Addison-Wesley Professional, 2002)
141. O. Etzion, P. Niblett, *Event Processing in Action* (Manning Publications, 2010)
142. P.T. Eugster, P.A. Felber, R. Guerraoui, A.-M. Kermarrec, The many faces of publish/subscribe. ACM Comput. Surv. **35**(2), 114–131 (2003)
143. G. Mühl, P. Pietzuch, L. Fiege, *Distributed Event-Based Systems* (2006)
144. S. Hasan, *Loose Coupling in Heterogeneous Event-Based Systems via Approximate Semantic Matching and Dynamic Enrichment* (National University of Ireland, Galway, 2016)
145. T. Heinze, Z. Jerzak, G. Hackenbroich, C. Fetzer, Latency-aware elastic scaling for distributed data stream processing systems, in *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems* (2014)
146. R. Tudoran et al., JetStream: enabling high performance event streaming across cloud datacenters, in *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems – DEBS '14* (2014)
147. B. Ottenwalder, B. Koldehofe, K. Rothermel, K. Hong, U. Ramachandran, RECEP: selection-based reuse for distributed complex event processing, in *DEBS 2014 – Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems* (2014)
148. N.K. Pandey, K. Zhang, S. Weiss, H.-A. Jacobsen, R. Vitenberg, Distributed event aggregation for content-based publish/subscribe systems, in *Proceedings of the 8th ACM International Conference on Distributed Event-Based Systems – DEBS '14* (2014)
149. T. Sakaki, M. Okazaki, Y. Matsuo, Earthquake shakes twitter users: real-time event detection by social sensors, in *Proceedings of the 19th International Conference on World Wide Web* (2010)
150. K.G. Shin, P. Ramanathan, Real-time computing: a new discipline of computer science and engineering. Proc. IEEE **82**(1), 6–24 (1994)
151. S. Hasan, E. Curry, Approximate semantic matching of events for the internet of things. ACM Trans. Internet Technol. **14**(1), 1–23 (2014)
152. L. Lefort, C. Henson, K. Taylor, P. Barnaghi, Semantic Sensor Network XG Final Report, W3C Incubator Group Report (28 June 2011)
153. K. Akpınar, K.A. Hua, K. Li, ThingStore: a platform for internet-of-things application development and deployment, in *Proceedings of the 9th ACM International Conference on Distributed and Event-Based Systems – DEBS '15* (2015), pp. 162–173
154. S. Lavalle, E. Lesser, R. Shockley, M.S. Hopkins, N. Kruschwitz, Big data, analytics and the path from insights to value. MIT Sloan Manag. Rev. **52**(2), 21–32 (2011)
155. A. Doan et al., Toward a system building agenda for data integration. IEEE Data Eng. Bull. **41** (1), 35–46 (2018)

156. U. ul Hassan et al., Water analytics and management with real-time linked dataspaces, in *Government 3.0 – Next Generation Government Technology Infrastructure and Services*, ed. by A. Ojo, J. Millar (Springer, 2017), pp. 173–196
157. E. Zaidi, G. De Simoni, R. Edjlali, A.D. Duncan, *Data Catalogs Are the New Black in Data Management and Analytics* (2017)
158. K. Bollacker, C. Evans, P. Paritosh, T. Sturge, J. Taylor, Freebase: a collaboratively created graph database for structuring human knowledge, in *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data – SIGMOD '08* (2008), pp. 1247–1250
159. C. Bizer et al., DBpedia – a crystallization point for the Web of Data. Web Semant. Sci. Serv. Agents World Wide Web **7**(3), 154–165 (2009)
160. N. Shadbolt, T. Berners-Lee, W. Hall, The semantic web revisited. IEEE Intell. Syst. **21**(3), 96–101 (2006)
161. E. Kaufmann, A. Bernstein, Evaluating the usability of natural language query languages and interfaces to Semantic Web knowledge bases. J. Web Semant. **8**(4), 377–393 (2010)
162. V. Kashyap, A. Sheth, Semantic and schematic similarities between database objects: a context-based approach. VLDB J. Int. J. Very Large Data Bases **5**(4), 276–304 (1996)
163. G.W. Furnas, T.K. Landauer, L.M. Gomez, S.T. Dumais, The vocabulary problem in human-system communication. Commun. ACM **30**(11), 964–971 (1987)
164. A. Hogan, A. Harth, J. Umbrich, S. Kinsella, A. Polleres, S. Decker, Searching and browsing linked data with SWSE: the semantic web search engine. J. Web Semant. **9**(4), 365–401 (2011)
165. R. Delbru, S. Campinas, G. Tummarello, Searching web data: an entity retrieval and high-performance indexing model. J. Web Semant. **10**, 33–58 (2012)
166. Q. Zhou, C. Wang, M. Xiong, H. Wang, Y. Yu, SPARK: adapting keyword query to semantic search, in *6th International The Semantic Web and 2Nd Asian Conference on Asian Semantic Web Conference, ISWC'07/ASWC'07* (2007), pp. 694–707
167. H. Wang et al., Semplore: a scalable IR approach to search the Web of Data. Web Semant. Sci. Serv. Agents World Wide Web **7**(3), 177–188 (2009)
168. X. Dong, A. Halevy, Indexing dataspaces, in *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data – SIGMOD '07* (2007), p. 43
169. V. Lopez, M. Fernández, E. Motta, N. Stieler, PowerAqua: supporting users in querying and exploring the Semantic Web. Semant. Web **3**(3), 249–265 (2012)
170. D. Damljanovic, M. Agatonovic, H. Cunningham, FREyA: an interactive way of querying linked data using natural language, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2012), pp. 125–138
171. A. Freitas, J. Oliveira, S. O'Riain, E. Curry, J.P. da Silva, Treo: best-effort natural language queries over linked data, in *Proceedings of the 16th International Conference on Applications of Natural Language to Information Systems, NLDB 2011 (poster)*, vol. 6716 (2011), pp. 286–289
172. C. Unger, L. Bühmann, J. Lehmann, A.-C. Ngonga Ngomo, D. Gerber, P. Cimiano, Template-based question answering over RDF data, in *Proceedings of the 21st International Conference on World Wide Web – WWW '12*, (2012)
173. A. Freitas, J. Oliveira, S. O'Riain, E. Curry, J.P. da Silva, Querying linked data using semantic relatedness: a vocabulary independent approach, in *Proceedings of the 16th International Conference on Applications of Natural Language to Information Systems, NLDB 2011*, vol. 6716 (2011), pp. 40–51
174. D. Ferrucci et al., Building Watson: an overview of the DeepQA project. *AI Magazine* (2010)
175. V. Sheokand, V. Singh, Best effort query answering in dataspaces on unstructured data, in *IEEE International Conference on Computing, Communication and Automation (ICCCA 2016)* (2017), pp. 155–159
176. N. Lal, S. Qamar, S. Shiwani, Search ranking for heterogeneous data over dataspace. Int. J. Control Theory Appl. **9**(20), 421–431 (2016)

177. N. Lal, S. Qamar, Comparison of ranking algorithms with dataspace, in *International Conference on Advances in Computer Engineering and Applications (ICACEA 2015)* (2015), pp. 565–572
178. M.A. Vaz Salles, J. Dittrich, L. Blunschi, Intensional associations in dataspaces, in *2010 IEEE 26th International Conference on Data Engineering (ICDE 2010)* (2010), pp. 984–987
179. M.A. Vaz Salles, J.-P. Dittrich, S.K. Karakashian, O.R. Girard, L. Blunschi, iTrails: pay-as-you-go information integration in dataspaces, in *VLDB 07 Proceedings of the 33rd International Conference on Very Large Data Bases* (2007), pp. 663–674
180. A. Das Sarma, X.L. Dong, A.Y. Halevy, Uncertainty in data integration and dataspace support platforms, in *Schema Matching and Mapping* (Springer, Berlin, 2011), pp. 75–108
181. S. Song, L. Chen, M. Yuan, Materialization and decomposition of dataspaces for efficient search. IEEE Trans. Knowl. Data Eng. **23**(12), 1872–1887 (2011)
182. S. Song, L. Chen, P.S. Yu, Comparable dependencies over heterogeneous data. VLDB J. **22** (2), 253–274 (2013)
183. P.D. Turney, P. Pantel, From frequency to meaning: vector space models of semantics. J. Artif. Intell. Res. (2010)
184. H. Lasi, P. Fettke, H.-G. Kemper, T. Feld, M. Hoffmann, Industry 4.0. Bus. Inf. Syst. Eng. **6** (4), 239–242 (2014)
185. J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of Things (IoT): a vision, architectural elements, and future directions. Futur. Gener. Comput. Syst. **29**(7), 1645–1660 (2013)
186. J. Manyika, M. Chui, J. Bughin, R. Dobbs, P. Bisson, A. Marrs, Disruptive technologies: advances that will transform life, business, and the global economy. *McKinsey Global Institute* (2013)
187. B. Ganter, R. Wille, *Formal Concept Analysis: Mathematical Foundations Formal Concept Analysis* (Springer, 1997)
188. W. Derguech, S. Bhiri, S. Hasan, E. Curry, Using formal concept analysis for organizing and discovering sensor capabilities. Comput. J. **58**(3), 356–367 (2015)
189. D. Elenius et al., The OWL-S Editor – a development tool for semantic web services, in *The Semantic Web: Research and Applications, Second European Semantic Web Conference, ESWC 2005*, vol. 3532 (2005), pp. 78–92
190. M. Aznag, M. Quafafou, Z. Jarir, Leveraging formal concept analysis with topic correlation for service clustering and discovery, in *Proceedings – 2014 IEEE International Conference on Web Services, ICWS 2014* (2014), pp. 153–160
191. S. Ben Mokhtar, D. Preuveneers, N. Georgantas, V. Issarny, Y. Berbers, EASY: Efficient semAntic Service discoverY in pervasive computing environments with QoS and context support. J. Syst. Softw. **81**(5), 785–808 (2008)
192. N. Srinivasan, M. Paolucci, K. Sycara, Adding OWL-S to UDDI, implementation and throughput, in *Proceedings of the 1st International Workshop on Semantic Web Services and Web Process Composition (SWSWPC 2004)* (2004), pp. 6–9
193. P. Oaks, A.H.M. ter Hofstede, D. Edmond, Capabilities: describing what services can do, in *International Conference on Service-Oriented Computing (ICSOC 2003)*, vol. 2910 (2003), pp. 1–16
194. A. Adala, N. Tabbane, S. Tabbane, A framework for automatic web service discovery based on semantics and NLP techniques. Adv. Multimed. **2011**, 1–7 (2011)
195. K.P. Sycara, S. Widoff, M. Klusch, J. Lu, Larks: dynamic matchmaking among heterogeneous software agents in cyberspace. Auton. Agent. Multi. Agent. Syst. **5**(2), 173–203 (2002)
196. D. Fensel, F.M. Facca, E. Simperl, I. Toma, Web service modeling ontology, in *Semantic Web Services* (Springer, Berlin, 2011), pp. 107–129
197. D. Martin et al., OWL-S: Semantic markup for web services. *W3C Member Submission* (2004), http://www.w3.org/Submission/OWL-S/
198. S. Ferndriger, A. Bernstein, J.S. Dong, Y. Feng, Y.-F. Li, J. Hunter, Enhancing semantic web services with inheritance, in *7th International Semantic Web Conference, (ISWC) 2008*. LNCS vol. 5318 (2008), pp. 162–177

199. L.A. Kamaruddin, J. Shen, G. Beydoun, Usage of OWL-S and WSMO in semantic web services projects: a survey, in *Eighth Asia-Pacific Conference on Conceptual Modelling 2012,* Melbourne, Australia, January 2012, **130**(1993), 53–58 (2012)
200. F. Strok, A. Neznanov, Comparing and analyzing the computational complexity of FCA algorithms, in *Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists on SAICSIT '10* (2010), pp. 417–420
201. United States Office of Management and Budget, *North American Industry Classification System (NAICS)* (2017)
202. United Nations Development Programme, *United Nations Standard Products and Services Code (UNSPSC)* (1998)
203. W. Binder, A.D. Mosincat, S. Spycher, I. Constantinescu, B. Faltings, Multiversion concurrency control for the generalized search tree. Concurr. Comput. Pract. Exp. **21**(12), 1547–1571 (2009)
204. J.M. Hellerstein, J.F. Naughton, A. Pfeffer, Generalized search trees for database systems, in *VLDB'95, Proceedings of 21th International Conference on Very Large Data Bases,* Zurich, Switzerland, 11–15 September 1995, pp. 562–573
205. T.W. Malone, K. Crowston, G.A. Herman, *Organizing Business Knowledge: The MIT Process Handbook*, vol. 1 (MIT Press, 2003)
206. R. Belohlávek, V. Vychodil, Background knowledge in formal concept analysis: constraints via closure operators, in *Proceedings of the 2010 ACM Symposium on Applied Computing (SAC)* (2010), pp. 1113–1114
207. J. Tane, T. Kaiser, S. Objedkov, *The Concept Explorer* (2012), http://conexp.sourceforge.net
208. S. Hasan, E. Curry, M. Banduk, S. O'Riain, Toward situation awareness for the semantic sensor web: complex event processing with dynamic linked data enrichment, in *4th International Workshop on Semantic Sensor Networks 2011 (SSN11)* (2011), pp. 69–81
209. G. Arévalo, S. Ducasse, O. Nierstrasz, Lessons learned in applying formal concept analysis to reverse engineering, in *Formal Concept Analysis, Third International Conference, (ICFCA) 2005*, vol. 3403, Lens, France, 14–18 February 2005, pp. 95–112
210. R. Godin, R. Missaoui, H. Alaoui, Incremental concept formation algorithms based on Galois (concept) lattices. Comput. Intell. **11**, 246–267 (1995)
211. A. Kittur et al., The future of crowd work, in *Proceedings of the 2013 conference on Computer supported cooperative work – CSCW '13* (2013), pp. 1301–1318
212. U. ul Hassan, E. Curry, A multi-armed bandit approach to online spatial task assignment, in *11th IEEE International Conference on Ubiquitous Intelligence and Computing (UIC 2014)* (2014)
213. U. ul Hassan, E. Curry, A capability requirements approach for predicting worker performance in crowdsourcing, in *9th IEEE International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom 2013)* (2013)
214. A. Doan, R. Ramakrishnan, A.Y. Halevy, Crowdsourcing systems on the World-Wide Web. Commun. ACM **54**(4), 86–96 (2011)
215. A.I. Chittilappilly, L. Chen, S. Amer-Yahia, A survey of general-purpose crowdsourcing techniques. IEEE Trans. Knowl. Data Eng. **28**(9), 2246–2266 (2016)
216. H. Garcia-Molina, M. Joglekar, A. Marcus, A. Parameswaran, V. Verroios, Challenges in data crowdsourcing. IEEE Trans. Knowl. Data Eng. **28**(4), 901–911 (2016)
217. G. Li, J. Wang, Y. Zheng, M.J. Franklin, Crowdsourced data management: a survey. IEEE Trans. Knowl. Data Eng. **28**(9), 2296–2319 (2016)
218. A.J. Quinn, B.B. Bederson, Human computation: a survey and taxonomy of a growing field, in *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems* (2011), pp. 1403–1412
219. E. Law, L. von Ahn, Human computation. Synth. Lect. Artif. Intell. Mach. Learn. **5**(3), 1–121 (2011)

220. D. Crowley, E. Curry, J. Breslin, Closing the loop – from citizen sensing to citizen actuation, in *7th IEEE International Conference on Digital Ecosystems and Technologies (IEEE DEST 2013)* (2013)
221. U. ul Hassan, A. Zaveri, E. Marx, E. Curry, J. Lehmann, ACRyLIQ: leveraging DBpedia for adaptive crowdsourcing in linked data quality assessment, in *20th International Conference on Knowledge Engineering and Knowledge Management (EKAW2016)* (2016)
222. G. Li, Y. Zheng, J. Fan, J. Wang, R. Cheng, Crowdsourced data management: overview and challenges, in *Proceedings of the 2017 ACM International Conference on Management of Data* (2017), pp. 1711–1716
223. A.G. Parameswaran, H. Park, H. Garcia-Molina, N. Polyzotis, J. Widom, Deco: declarative crowdsourcing, in *Proceedings of the 21st ACM International Conference on Information and Knowledge Management – CIKM '12* (2012), p. 1203
224. A.P. Kulkarni, M. Can, B. Hartmann, Turkomatic: automatic recursive task and workflow design for mechanical turk, in *Proceedings of the 2011 Annual Conference Extended Abstracts on Human Factors in Computing Systems – CHI EA '11* (2011), p. 2053
225. E. Law, M. Yin, J. Goh, K. Chen, M.A. Terry, K.Z. Gajos, Curiosity killed the cat, but makes crowdwork better, in *Proc. 2016 CHI Conference on Human Factors in Computing Systems – CHI '16* (2016), pp. 4098–4110
226. B. Bergvall-Kåreborn, D. Howcroft, Amazon Mechanical Turk and the commodification of labour. New Technol. Work Employ. **29**(3), 213–223 (2014)
227. R.M. Borromeo, T. Laurent, M. Toyama, The influence of crowd type and task complexity on crowdsourced work quality, in *Proceedings of the 20th International Database Engineering & Applications Symposium on – IDEAS '16* (2016), pp. 70–76
228. U. ul Hassan, S. O'Riain, E. Curry, Towards expertise modelling for routing data cleaning tasks within a community of knowledge workers, in *17th International Conference on Information Quality (ICIQ 2012)* (2012)
229. S. Basu Roy, I. Lykourentzou, S. Thirumuruganathan, S. Amer-Yahia, G. Das, Task assignment optimization in knowledge-intensive crowdsourcing. VLDB J. **24**(4), 467–491 (2015)
230. A. Slivkins, J.W. Vaughan, Online decision making in crowdsourcing markets. ACM SIGecom Exch. **12**(2), 4–23 (2014)
231. F. Rodrigues, F. Pereira, B. Ribeiro, Learning from multiple annotators: distinguishing good from random labelers. Pattern Recognit. Lett. **34**(12), 1428–1436 (2013)
232. F. Ma et al., FaitCrowd: fine grained truth discovery for crowdsourced data aggregation, in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (2015), pp. 745–754,
233. R. Teodoro, P. Ozturk, M. Naaman, W. Mason, J. Lindqvist, The motivations and experiences of the on-demand mobile workforce, in *Proceedings of the 17th ACM Conference on Computer Supported Cooperative Work & Social Computing – CSCW '14* (2014), pp. 236–247
234. B. Morschheuser, J. Hamari, J. Koivisto, Gamification in crowdsourcing: a review. Proc. Annu. Hawaii Int. Conf. Syst. Sci. **March**, 4375–4384 (2016)
235. J. Goncalves et al., Crowdsourcing on the spot: altruistic use of public displays, feasibility, performance, and behaviours, in *Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing* (2013), pp. 753–762
236. D. Haas, J. Wang, E. Wu, M.J. Franklin, CLAMShell: speeding up crowds for low-latency data labeling. Proc. VLDB Endow. **9**(4), 372–383 (2015)
237. J. Fan, M. Zhang, S. Kok, M. Lu, B.C. Ooi, CrowdOp: query optimization for declarative crowdsourcing systems. IEEE Trans. Knowl. Data Eng. **27**(8), 2078–2092 (2015)
238. V. Verroios, P. Lofgren, H. Garcia-Molina, tDP: an optimal-latency budget allocation strategy for crowdsourced MAXIMUM operations, in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data – SIGMOD '15* (2015), pp. 1047–1062
239. M.S. Bernstein, J. Brandt, R.C. Miller, D.R. Karger, Crowds in two seconds: enabling realtime crowd-powered interfaces, in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (2011), pp. 33–42

240. J.P. Bigham et al., VizWiz: nearly real-time answers to visual questions, in *Proceedings of the 23nd Annual ACM Symposium on User Interface Software and Technology* (2010), pp. 333–342
241. P. Yadav, U. ul Hassan, S. Hasan, E. Curry, The event crowd: a novel approach for crowd-enabled event processing, in *The 11th ACM International Conference on Distributed and Event-Based Systems (DEBS '17)* (2017)
242. M. Sabou, K. Bontcheva, A. Scharl, Crowdsourcing research opportunities: lessons from natural language processing, in *Proceedings of the 12th International Conference on Knowledge Management and Knowledge Technologies* (2012), pp. 17:1–17:8
243. K. Belhajjame, N.W. Paton, S.M. Embury, A.A.A. Fernandes, C. Hedeler, Feedback-based annotation, selection and refinement of schema mappings for dataspaces, in *Proceedings of the 13th International Conference on Extending Database Technology – EDBT '10* (2010), p. 573
244. Y. Qi, K.S. Candan, M.L. Sapino, FICSR: feedback-based inconsistency resolution and query processing on misaligned data sources, in *Proceedings of the 2007 ACM SIGMOD International Conference on Management of Data – SIGMOD '07* (2007), p. 151
245. M. van Keulen, A. de Keijzer, Qualitative effects of knowledge rules and user feedback in probabilistic data integration. VLDB J. **18**(5), 1191–1217 (2009)
246. M. Yakout, A.K. Elmagarmid, J. Neville, M. Ouzzani, I.F. Ilyas, Guided data repair. Proc. VLDB Endow. **4**(5), 279–289 (2011)
247. X. Chu et al., KATARA: a data cleaning system powered by knowledge bases and crowdsourcing, in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data* (2015), pp. 1247–1261
248. J. Zhou et al., CrowdAidRepair: a crowd-aided interactive data repairing method, in *International Conference on Database Systems for Advanced Applications (DASFAA 2016)*, vol. 9642 (2016), pp. 51–66
249. A. Marcus, E. Wu, D. Karger, S. Madden, R. Miller, Human-powered sorts and joins. Proc. VLDB Endow. **5**(1), 13–24 (2011)
250. M.J. Franklin, D. Kossmann, T. Kraska, S. Ramesh, R. Xin, CrowdDB: answering queries with crowdsourcing, in *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data* (2011), pp. 61–72
251. X. Chai, B. Vuong, A. Doan, J.F. Naughton, Efficiently incorporating user feedback into information extraction and integration programs, in *Proceedings of the 35th SIGMOD International Conference on Management of Data – SIGMOD '09* (2009), p. 87
252. S. Kochhar, S. Mazzocchi, P. Paritosh, The anatomy of a large-scale human computation engine, in *Proceedings of the ACM SIGKDD Workshop on Human Computation – HCOMP '10* (2010), pp. 10–17
253. P. Buneman, J. Cheney, S. Lindley, H. Mueller, The database Wiki project: a general-purpose platform for data curation and collaboration. ACM SIGMOD Rec. **40**(3), 15–20 (2011)
254. R. McCann, W. Shen, A. Doan, Matching schemas in online communities: a Web 2.0 approach, in *2008 IEEE 24th International Conference on Data Engineering* (2008), pp. 110–119
255. U. ul Hassan, E. Curry, Efficient task assignment for spatial crowdsourcing: a combinatorial fractional optimization approach with semi-bandit learning. Expert Syst. Appl. **58**, 36–56 (2016)
256. U. ul Hassan, M. Bassora, A.H. Vahid, S. O'Riain, E. Curry, A collaborative approach for metadata management for internet of things: linking micro tasks with physical objects, in *First International Workshop on Internet of Things* (2013)
257. J.M. Mortensen, M.A. Musen, N.F. Noy, Crowdsourcing the verification of relationships in biomedical ontologies, in *AMIA ... Annual Symposium Proceedings. AMIA Symposium* (2013), pp. 1020–1029
258. G. Demartini, D.E. Difallah, P. Cudré-Mauroux, Large-scale linked data integration using probabilistic reasoning and crowdsourcing. VLDB J. **22**(5), 665–687 (2013)

259. C. Gokhale et al., Corleone: hands-off crowdsourcing for entity matching, in *ACM SIGMOD International Conference on Management of Data – SIGMOD '14* (2014), pp. 601–612
260. Y. Zhuang, G. Li, Z. Zhong, J. Feng, C. Science, Hike: a hybrid human-machine method for entity alignment in large-scale knowledge bases. CIKM, 1917–1926 (2017)
261. J. Wang, T. Kraska, M.J. Franklin, J. Feng, CrowdER: crowdsourcing entity resolution. Proc. VLDB Endow. **5**(11), 1483–1494 (2012)
262. J. Lee, D. Lee, S. Kim, CrowdSky: skyline computation with crowdsourcing, in *Proceedings of 19th International Conference on Extending Database Technology* (2016), pp. 125–136
263. Y. Tong, C.C. Cao, C.J. Zhang, Y. Li, L. Chen, CrowdCleaner: data cleaning for multi-version data on the web via crowdsourcing, in *2014 IEEE 30th International Conference on Data Engineering* (2014), pp. 1182–1185
264. E. Kamar, S. Hacker, E. Horvitz, Combining human and machine intelligence in large-scale crowdsourcing, in *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, vol. 1 (2012), pp. 467–474
265. D. Crowley, E. Curry, J. Breslin, Citizen actuation for smart environments. IEEE Consum. Electron. Mag. **5**(3), 90–94 (2016)
266. U. ul Hassan, S. O'Riain, E. Curry, SLUA: towards semantic linking of users with actions in crowdsourcing, in *International Workshop on Crowdsourcing the Semantic Web* (2013)
267. L. Li, W. Chu, J. Langford, R.E. Schapire, A contextual-bandit approach to personalized news article recommendation, in *Proceedings of the 19th International Conference on World Wide Web – WWW '10* (2010), p. 661
268. N. Marz, J. Warren, *Big Data, Principles and Best Practices of Scalable Real-Time Data Systems* (Manning Publications, 2015)
269. R.B. Miller, Response time in man-computer conversational transactions, in *Proceedings of the Fall Joint Computer Conference, Part I on – AFIPS '68 (Fall, Part I)*, 9–11 December 1968
270. J. Nielsen, *Usability Engineering* (Morgan Kaufmann, 1993)
271. A. Hinze, K. Sachs, A. Buchmann, Event-based applications and enabling technologies, in *Proceedings of the Third ACM International Conference on Distributed Event-Based Systems – DEBS '09* (2009), p. 1
272. S. Hasan, E. Curry, Thematic event processing, in *Proceedings of the 15th International Middleware Conference on – Middleware '14* (2014), pp. 109–120
273. Q. Wu, Q. Zhu, X. Jian, QoS-aware multi-granularity service composition based on generalized component services, in *International Conference on Service-Oriented Computing (ICSOC2013)*. LNCS vol. 8274 (2013), pp. 446–455
274. F. Gao, A. Intizar, E. Curry, A. Mileo, QoS-aware stream federation and optimization based on service composition. Int. J. Semant. Web Inf. Syst. (2016)
275. F. Gao, E. Curry, S. Bhiri, Complex event service provision and composition based on event pattern matchmaking, in *8th ACM International Conference on Distributed Event-Based Systems (DEBS 2014)* (2014), pp. 71–82
276. F. Gao, M.I. Ali, E. Curry, A. Mileo, Automated discovery and integration of semantic urban data streams: the ACEIS middleware. Futur. Gener. Comput. Syst. **76**, 561–581 (2017)
277. M.C. Jaeger, G. Rojec-Goldmann, G. Muhl, QoS aggregation for web service composition using workflow patterns, in *Proceedings of the Eighth IEEE International Enterprise Distributed Object Computing Conference, 2004. EDOC 2004* (2004), pp. 149–159
278. C. Gao, M. Cai, H. Chen, QoS-aware service composition based on tree-coded genetic algorithm. Proc. Int. Comput. Softw. Appl. Conf. **1**, 361–367 (2007)
279. F. Karatas, D. Kesdogan, An approach for compliance-aware service selection with genetic algorithms, in *International Conference on Service-Oriented Computing (ICSOC 2013)*. LNCS vol. 8274 (2013), pp. 465–473
280. Z. Laliwala, S. Chaudhary, Event-driven service-oriented architecture, in *5th International Conference Service Systems and Service Management – Exploring Service Dynamics with Science and Innovative Technology, ICSSSM'08* (2008)

281. D.B.D. Bo, D.K.D. Kun, Z.X.Z. Xiaoyi, A high performance enterprise service bus platform for complex event processing, in *2008 Seventh International Conference on Grid and Cooperative Computing* (2008), pp. 578–583
282. J. Schiefer, S. Rozsnyai, C. Rauscher, G. Saurer, Event-driven rules for sensing and responding to business situations, in *Proceedings of the 2007 Inaugural International Conference on Distributed Event-Based Systems – DEBS '07* (2007), p. 198
283. D. Dell'Aglio, J.-P. Calbimonte, E. Della Valle, O. Corcho, Towards a unified language for RDF stream query processing, in *Revised Selected Papers of the ESWC 2015 Satellite Events on The Semantic Web: ESWC 2015 Satellite Events*, vol. 9341 (2015), pp. 353–363
284. D. Zimmer, R. Unland, On the semantics of complex events in active database management systems, in *Proceedings 15th International Conference on Data Engineering (Cat. No. 99CB36337)*, no. Dml (1999), pp. 392–399
285. K. Whitehouse, F. Zhao, J. Liu, Semantic streams: a framework for composable semantic interpretation of sensor data, in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. LNCS vol. 3868 (2006), pp. 5–20
286. Q. Zhou, Y. Simmhan, V. Prasanna, Towards hybrid online on-demand querying of realtime data with stateful complex event processing, in *Proceedings – 2013 IEEE International Conference on Big Data, Big Data 2013* (2013), pp. 199–205
287. F. Heintz, DyKnow: a stream-based knowledge processing middleware framework (2009)
288. Y. Qin, Q.Z. Sheng, N.J.G. Falkner, S. Dustdar, H. Wang, A.V. Vasilakos, When things matter: a survey on data-centric internet of things. J. Netw. Comput. Appl. **64**, 137–153 (2016)
289. J. Pan, R. Jain, S. Paul, T. Vu, A. Saifullah, M. Sha, An internet of things framework for smart energy in buildings: designs, prototype, and experiments. IEEE Internet Things J. (2015)
290. J.G.J. Gao, L. Guibas, N. Milosavljevic, D.Z.D. Zhou, Distributed resource management and matching in sensor networks, in *2009 International Conference on Information Processing in Sensor Networks* (2009)
291. S. Mathur et al., ParkNet: drive-by sensing of road-side parking statistics, in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services – MobiSys '10* (2010), p. 123
292. A. James, J. Cooper, K. Jeffery, G. Saake, *Research Directions in Database Architectures for the Internet of Things: A Communication of the First International Workshop on Database Architectures for the Internet of Things (DAIT 2009)* (Springer, Berlin, 2009), pp. 225–233
293. P. Barnaghi, A. Sheth, C. Henson, From data to actionable knowledge: big data challenges in the web of things [Guest Editors' Introduction]. IEEE Intell. Syst. **28**(6), 6–11 (2013)
294. P. Barnaghi, W. Wang, C. Henson, K. Taylor, Semantics for the internet of things. Int. J. Semant. Web Inf. Syst. **8**(1), 1–21 (2012)
295. Y. Qin, Q.Z. Sheng, E. Curry, Matching over linked data streams in the internet of things. IEEE Internet Comput. **19**(3), 21–27 (2015)
296. Y. Qin, Q.Z. Sheng, N.J.G. Falkner, A. Shemshadi, E. Curry, Batch matching of conjunctive triple patterns over linked data streams in the internet of things, in *27th International Conference on Scientific and Statistical Database Management (SSDBM 2015)* (2015)
297. H. Haverkort, An inventory of three-dimensional Hilbert space-filling curves. CoRR (2011)
298. D. Le-Phuoc, M. Dao-Tran, J. Xavier Parreira, M. Hauswirth, A native and adaptive approach for unified processing of linked streams and linked data, in *International Semantic Web Conferenec (SWC 2011)* (Springer, 2011), pp. 370–388
299. D. Anicic, P. Fodor, S. Rudolph, N. Stojanovic, EP-SPARQL: a unified language for event processing and stream reasoning, in *Proceedings of the 20th International Conference on World Wide Web* (2011), pp. 635–644
300. A. Harth, K. Hose, M. Karnstedt, A. Polleres, K.-U. Sattler, J. Umbrich, Data summaries for on-demand queries over linked data, in *Proceedings of the 19th International Conference on World Wide Web – WWW '10* (2010)

301. Y. Diao, M. Altinel, M.J. Franklin, H. Zhang, P. Fischer, Path sharing and predicate evaluation for high-performance XML filtering. ACM Trans. Database Syst. (2003)
302. T. Imielinski, S. Viswanathan, B.R. Badrinath, Data on air: organization and access. IEEE Trans. Knowl. Data Eng. **9**(3), 353–372 (1997)
303. J. Umbrich, K. Hose, M. Karnstedt, A. Harth, A. Polleres, Comparing data summaries for processing live queries over linked data. World Wide Web **14**(5–6), 495–544 (2011)
304. A. Guttman, R-trees: a dynamic index structure for spatial searching, in *Proceedings of the 1984 ACM SIGMOD International Conference on Management of Data – SIGMOD '84* (1984), p. 47
305. B. Zheng, W.-C. Lee, D. Lun Lee, On searching continuous k nearest neighbors in wireless data broadcast systems. IEEE Trans. Mob. Comput. **6**(7), 748–761 (2007)
306. S. Petrovic, M. Osborne, V. Lavrenko, Streaming first story detection with application to Twitter, in *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (2010)
307. S. Hasan, S. O'Riain, E. Curry, Approximate semantic matching of heterogeneous events, in *6th ACM International Conference on Distributed Event-Based Systems (DEBS 2012)* (2012), pp. 252–263
308. S. Hasan, E. Curry, Thingsonomy: tackling variety in internet of things events. IEEE Internet Comput. **19**(2), 10–18 (2015)
309. E. Gabrilovich, S. Markovitch, Computing semantic relatedness using wikipedia-based explicit semantic analysis, in *Proceedings of the 20th International Joint Conference on Artifical Intelligence* (2007), pp. 1606–1611
310. M. Gupta, R. Li, Z. Yin, J. Han, Survey on social tagging techniques. ACM SIGKDD Explor. Newsl. **12**(1), 58–72 (2010)
311. Z. Bellahsene, A. Bonifati, F. Duchateau, Y. Velegrakis, On evaluating schema matching and mapping, in *Schema Matching and Mapping* (Springer, Berlin, 2011), pp. 253–291
312. Merriam-Webster's, *Merriam-Webster Online Dictionary*
313. D. Carvalho, C. Çalli, A. Freitas, E. Curry, EasyESA: a low-effort infrastructure for explicit semantic analysis (demonstration paper in proceedings), in *Proceedings of the 13th International Semantic Web Conference (ISWC 2014)* (2014)
314. A. Carzaniga, D.S. Rosenblum, A.L. Wolf, Achieving scalability and expressiveness in an internet-scale event notification service, in *Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing* (2000), pp. 219–227
315. P.T. Eugster, R. Guerraoui, C.H. Damm, On objects and events. ACM SIGPLAN Not. **36**(11), 254–269 (2001)
316. L. Fiege, M. Cilia, G. Muhl, A. Buchmann, Publish-subscribe grows up: support for management, visibility control, and heterogeneity. IEEE Internet Comput. **10**(1), 48–55 (2006)
317. M. Petrovic, I. Burcea, H.-A. Jacobsen, S-ToPSS: semantic Toronto publish/subscribe system, in *29th International Conference on Very Large Data Bases*, vol. 29 (2003), pp. 1101–1104
318. J. Wang, B. Jin, J. Li, An ontology-based publish/subscribe system, in *Middleware 2004*, vol. 2002 (2004), pp. 232–253
319. L. Zeng, H. Lei, A semantic publish/subscribe system, in *E-Commerce Technology for Dynamic E-Business, 2004. IEEE International Conference on* (2004), pp. 32–39
320. G.S. Blair et al., The role of ontologies in emergent Middleware: supporting interoperability in complex distributed systems, in *Middleware 2011* (Springer, 2011), pp. 410–430
321. W. Zhang, J. Ma, D. Ye, FOMatch: a fuzzy ontology-based semantic matching algorithm of publish/subscribe systems, in *Computational Intelligence for Modelling Control & Automation, 2008 International Conference on* (2008), pp. 111–117
322. H. Liu, H.-A. Jacobsen, Modeling uncertainties in publish/subscribe systems, in *Proceedings of the 20th International Conference on Data Engineering* (2004), pp. 510–521
323. H. Liu, H.-A. Jacobsen, A-ToPSS: a publish/subscribe system supporting imperfect information processing, in *Thirtieth International Conference on Very Large Data Bases*, vol. 30 (2004), pp. 1281–1284

324. S. Wasserkrug, A. Gal, O. Etzion, Y. Turchin, Efficient processing of uncertain events in rule-based systems. Knowl. Data Eng. IEEE Trans. **24**(1), 45–58 (2012)
325. A. Arasu, S. Babu, J. Widom, The CQL continuous query language: semantic foundations and query execution. VLDB J. **15**(2), 121–142 (2006)
326. K. Teymourian, M. Rohde, A. Paschke, Fusion of background knowledge and streams of events, in *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems* (2012), pp. 302–313
327. T. Freudenreich, S. Appel, S. Frischbier, A.P. Buchmann, ACTrESS, in *Proceedings of the 6th ACM International Conference on Distributed Event-Based Systems – DEBS '12* (2012), pp. 179–190
328. M. Cilia, C. Bornhövd, A.P. Buchmann, CREAM: an infrastructure for distributed, heterogeneous event-based applications, in *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE* (Springer, 2003), pp. 482–502
329. M. Cilia, M. Antollini, C. Bornhövd, A. Buchmann, Dealing with heterogeneous data in pub/sub systems: the concept-based approach, in *Proceedings Distributed Event Based Systems* (2004), pp. 26–31
330. F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the internet of things, in *Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing – MCC '12* (2012), p. 13
331. M. Gupta, J. Han, Heterogeneous network-based trust analysis. ACM SIGKDD Explor. Newsl. **13**(1), 54 (2011)
332. M.S. Felix Naumann, J.C. Freytag, Quality-driven source selection using data envelopment analysis, in *Third Conference on Information Quality* (1998)
333. J.O. Kephart, D.M. Chess, The vision of autonomic computing. Computer (Long. Beach. Calif.) **36**(1), 41–50 (2003)
334. P. Horn, Autonomic computing: IBM's perspective on the state of information technology. IBM Corp. Tech. Rep. (2001)
335. W. Derguech, E. Bruke, E. Curry, An autonomic approach to real-time predictive analytics using open data and the web of things, in *The 11th IEEE International Conference on Ubiquitous Intelligence and Computing (UIC 2014)* (2014)
336. J.R. McBride, A.R. Lupini, M.A. Schreuder, N.J. Smith, S.J. Pennycook, S.J. Rosenthal, An architectural blueprint for autonomic computing. *IBM Whitepaper* (2005)
337. M.J. Ismail, R. Ibrahim, I. Ismail, Adaptive neural network prediction model for energy consumption, in *2011 3rd International Conference on Computer Research and Development* (2011), pp. 109–113
338. T. Joachims, Making large-scale SVM learning practical, in *Advances in Kernel Methods – Support Vector Learning* (1999)
339. X. Wang, P. Guo, X. Huang, A review of wind power forecasting models. Energy Procedia **12**, 770–778 (2011)
340. M.A. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update. SIGKDD Explor. (2009)
341. J. Boyd, A discourse on winning and losing. *Air Univ. Doc. MU43947, briefing*, vol. 1 (1987)
342. J. Froehlich, L. Findlater, J. Landay, The design of eco-feedback technology, in *Proceedings of the 28th International Conference on Human Factors in Computing Systems – CHI '10* (2010)
343. J.O. Prochaska, C.C. DiClemente, Transtheoretical therapy: Toward a more integrative model of change. Psychother. Theory, Res. Pract. **19**(3), 276–288 (1982)
344. J.O. Prochaska, W.F. Velicer, The transtheoretical model of health behavior change. Am. J. Heal. Promot. **12**(1), 38–48 (1997)
345. H. He, S. Greenberg, E. Huang, One size does not fit all: applying the transtheoretical model to energy feedback technology design, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI '10* (2010)

346. E.S. Geller, The challenge of increasing proenvironment behavior. Handb. Environ. Psychol. (2002)
347. B.J. Fogg, C. Soohoo, D.R. Danielson, L. Marable, J. Stanford, E.R. Tauber, How do users evaluate the credibility of Web sites?, in *Proceedings of the 2003 Conference on Designing for User Experiences – DUX '03* (2003)
348. D. Lehrer, J. Vasudev, Visualizing energy information in commercial buildings: a study of tools, expert users, and building occupants (2011)
349. D. Lehrer, J. Vasudev, Visualizing information to improve building performance: a study of expert users, in *ACEEE Summer Study on Energy Efficiency in Buildings* (2010)
350. B. Johnson, B. Shneiderman, Tree-Maps: a space-filling approach to the visualization of hierarchical information structures, in *Proceedings of the 2nd Conference on Visualization '91* (1991), pp. 284–291
351. B. Craft, P. Cairns, Beyond guidelines: what can we learn from the visual information seeking mantra?, in *Proceedings of the International Conference on Information Visualisation* (2005)
352. R.B. Cialdini, *Influence: Science and Practice* (2001)
353. A. Laskey, O. Kavazovic, OPower: energy efficiency through behavioral science and technology. XRDS Crossroads, ACM Mag. Students **17**(4), 47 (2011)
354. L. Festinger, *A Theory of Cognitive Dissonance* (Stanford University Press, 1957)
355. R.M. Ryan, E.L. Deci, Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. Am. Psychol. **55**(1), 68–78 (2000)
356. G. Mancero, W. Wong, P. Amaldi, Looking but not seeing: implications for HCI, in *Proceedings of the 14th European Conference on Cognitive Ergonomics Invent! Explore! – ECCE '07* (2007), p. 167
357. W. Abrahamse, L. Steg, Social influence approaches to encourage resource conservation: a meta-analysis. Glob. Environ. Chang. **23**(6), 1773–1785 (2013)
358. P.C. Stern, New environmental theories: toward a coherent theory of environmentally significant behavior. J. Soc. Issues **56**(3), 407–424 (2000)
359. K. Werbach, D. Hunter, *For the Win: How Game Thinking Can Revolutionize Your Business* (Wharton, 2012)
360. G. Zichermann, C. Cunningham, *Gamification by Design: Implementing Game Mechanics in Web and Mobile Apps*, 1st edn. (O'Reilly Media, 2001)
361. L.J. Becker, Joint effect of feedback and goal setting on performance: a field study of residential energy conservation. J. Appl. Psychol. **63**(4), 428–433 (1978)
362. J.H. van Houwelingen, W.F. van Raaij, The effect of goal-setting and daily electronic feedback on in-home energy use. J. Consum. Res. **16**(1), 98 (1989)
363. M.H. Gonzales, E. Aronson, M.A. Costanzo, Using social cognition and persuasion to promote energy conservation: a quasi-experiment. J. Appl. Soc. Psychol. **18**(12), 1049–1066 (1988)
364. T.H. Wang, R.D. Katzev, Group commitment and resource conservation: two field experiments on promoting recycling. J. Appl. Soc. Psychol. (1990)
365. M.S. Pallak, W. Cummings, Commitment and voluntary energy conservation. Personal. Soc. Psychol. Bull. **2**(1), 27–30 (1976)
366. C. Mercier, L. Moorefield, Commercial office plug load savings and assessment: final report. Calif. Energy Comm. (2011)
367. P.M. Gollwitzer, Implementation intentions: strong effects of simple plans. Am. Psychol. **54** (7), 493–503 (1999)
368. S. Bamberg, Effects of implementation intentions on the actual performance of new environmentally friendly behaviours — results of two field experiments. J. Environ. Psychol. **22**(4), 399–411 (2002)
369. D. Foster, S. Lawson, J. Wardman, M. Blythe, C. Linehan, 'Watts in it for me?': design implications for implementing effective energy interventions in organisations, in *Proceedings of the 2012 ACM Annual Conference on Human Factors in Computing Systems – CHI '12* (2012), p. 2357

370. F.W. Siero, A.B. Bakker, G.B. Dekker, M.T.C. Van Den Burg, Changing organizational energy consumption behaviour through comparative feedback. J. Environ. Psychol. **16**(3), 235–246 (1996)
371. R. Bartle, Hearts, clubs, diamonds, spades: players who suit MUDs. J. MUD Res. **1**(1), 19 (1996)
372. B.F. Skinner, *Science and Human Behavior* (Simon and Schuster, 1953)
373. B.J. Fogg, Persuasive technology: using computers to change what we think and do. Ubiquity **2002**(December), 2 (2002)
374. A.J. Resnik, R.B. Cialdini, Influence: science & practice. J. Mark. Res. **23**(3), 305 (1986)
375. B. Buxton, *Sketching User Experiences* (Elsevier, 2007)
376. A.R. Carrico, M. Riemer, Motivating energy conservation in the workplace: an evaluation of the use of group-level feedback and peer education. J. Environ. Psychol. **31**(1), 1–13 (2011)
377. A. Sheth, Citizen sensing, social signals, and enriching human experience. IEEE Internet Comput. **13**(4), 87–92 (2009)
378. A.K. Przybylski, K. Murayama, C.R. DeHaan, V. Gladwell, Motivational, emotional, and behavioral correlates of fear of missing out. Comput. Human Behav. **29**(4), 1841–1848 (2013)
379. J. Ho, S.S. Intille, Using context-aware computing to reduce the perceived burden of interruptions from mobile devices, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems – CHI '05* (2005)
380. A. Ojo, E. Curry, F. Sanaz-Ahmadi, A tale of open data innovations in five smart cities, in *48th Annual Hawaii International Conference on System Sciences (HICSS-48)* (2015), pp. 2326–2335
381. M. van Steen, G. Pierre, S. Voulgaris, Challenges in very large distributed systems. J. Internet Serv. Appl. **3**(1), 59–66 (2011)
382. V. Issarny et al., Service-oriented middleware for the future internet: state of the art and research directions. J. Internet Serv. Appl. **2**(1), 23–45 (2011)
383. C.N. Höfer, G. Karagiannis, Cloud computing services: taxonomy and comparison. J. Internet Serv. Appl. **2**(2), 81–94 (2011)
384. B. Billet, V. Issarny, Dioptase: a distributed data streaming middleware for the future web of things. J. Internet Serv. Appl. **5**(1), 1–19 (2014)
385. A. Ojo, E. Curry, T. Janowski, Designing next generation smart city initiatives – harbessing findings and lessons from a study of 10 smart city programs, in *22nd European Conference on Information Systems (ECIS 2014)*, vol. 2050 (2014), pp. 1–14
386. J.M. Schleicher et al., A holistic, interdisciplinary decision support system for sustainable smart city design, in *International Conference on Smart Cities (SMART-CT 2016)* (2016)
387. E.M. Rogers, *Diffusion of Innovations* (Free Press, 1962)
388. P. Anderson, M.L. Tushman, Managing through cycles of technological change. Res. Technol. Manag. **34**(3), 26–31 (1991)
389. J.P. Wisdom, K.H.B. Chor, K.E. Hoagwood, S.M. Horwitz, Innovation adoption: a review of theories and constructs. Adm. Policy Ment. Heal. Ment. Heal. Serv. Res. **41**(4), 480–502 (2014)
390. A. Freitas, E. Curry, Natural language queries over heterogeneous linked data graphs: a distributional-compositional semantics approach, in *18th International Conference on Intelligent User Interfaces (IUI'14)* (2014), pp. 279–288
391. X.-Q. Dong, B. Guo, Y. Shen, X.-L. Duan, Y.-C. Shen, H. Zhang, An efficient and secure decentralizing data sharing model. Jisuanji Xuebao/Chinese J. Comput. **41**(5), 1021–1036 (2018)
392. M. Langheinrich, Privacy by design—principles of privacy-aware ubiquitous systems, in *International conference on Ubiquitous Computing* (2001), pp. 273–291
393. J.A. Buck, S. Villines, We the people: consenting to a deeper democracy: a guide to sociocratic principles and methods. *Sociocracy.info* (2007)
394. B. Cohen, Incentives build robustness in BitTorrent, in *Workshop on Economics of Peer-to-Peer systems*, vol. 6 (2003), 68–72

395. J. Pouwelse, P. Garbacki, D. Epema, H. Sips, The BitTorrent P2P file-sharing system: measurements and analysis, in *4th International Conference on Peer-to-Peer Systems* (2005), pp. 205–216

396. A. Freitas, T. Knap, S. O'Riain, E. Curry, W3P: building an OPM based provenance model for the web. Futur. Gener. Comput. Syst. **27**(6), 766–774 (2011)