ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

Scuola di Ingegneria e Architettura
Corso di Laurea Magistrale in Ingegneria e Scienze Informatiche

A SEMANTIC WEB APPROACH TO ONTOLOGY-BASED SYSTEM: INTEGRATING, SHARING AND ANALYSING IOT HEALTH AND FITNESS DATA

Elaborata nel corso di: Web Semantico

Tesi di Laurea di:
ROBERTO REDA

Relatore:
Chiar.ma Prof.ssa
ANTONELLA CARBONARO

Correlatore:
Dott. FILIPPO PICCININI

ANNO ACCADEMICO 2016–2017
SESSIONE II

KEYWORDS

Semantic Web
Healthcare
Ontology
Internet of Things
Cognitive Computing

Abstract

With the rapid development of fitness industry, Internet of Things (IoT) technology is becoming one of the most popular trends for the health and fitness areas. IoT technologies have revolutionised the fitness and the sport industry by giving users the ability to monitor their health status and keep track of their training sessions. More and more sophisticated wearable devices, fitness trackers, smart watches and health mobile applications will appear in the near future. These systems do collect data non-stop from sensors and upload them to the Cloud. However, from a data-centric perspective the landscape of IoT fitness devices and wellness appliances is characterised by a plethora of representation and serialisation formats. The high heterogeneity of IoT data representations and the lack of common accepted standards, keep data isolated within each single system, preventing users and health professionals from having an integrated view of the various information collected. Moreover, in order to fully exploit the potential of the large amounts of data, it is also necessary to enable advanced analytics over it, thus achieving actionable knowledge. Therefore, due the above situation, the aim of this thesis project is to design and implement an ontology based system to (1) allow data interoperability among heterogeneous IoT fitness and wellness devices, (2) facilitate the integration and the sharing of information and (3) enable advanced analytics over the collected data (Cognitive Computing). The novelty of the proposed solution lies in exploiting Semantic Web technologies to formally describe the meaning of the data collected by the IoT devices and define a common communication strategy for information representation and exchange.

Sommario (Italian Abstract)

Con il rapido sviluppo dell'industria del fitness, la tecnologia delle Internet of Things (IoT) sta diventando una tra le più popolari nell'area della salute e dello sport. Le tecnologie IoT hanno rivoluzionato l'industria del fitness e dello sport fornendo agli utenti la possibilità di monitorare il loro stato di salute e tener traccia delle loro sessioni di allenamento. In avvenire continueranno ad apparire dispositivi indossabili, fitness trackers, e smart watches sempre più sofisticati. Questi sistemi acquisiscono dati dai sensori su base regolare e continuativa, rendendoli disponibili nel Cloud. Tuttavia, da un punto di vista datacentrico il panorama dei dispositivi IoT per il fitness e il wellness è caratterizzato da una moltitudine di formati di rappresentazione e serializzazione. La vasta eterogeneità di rappresentazione dei dati IoT e la mancanza di uno standard di riferimento comune, isola i dati all'interno di ciascun singolo sistema, privando gli utenti e i professionisti della salute di una vista integrata delle varie informazioni raccolte. Inoltre, per sfruttare a pieno il potenziale dell'enorme quantitativo di dati è necessario consentire tecniche avanzate di analisi così da poter estrarre conoscenza utile e significativa. Perciò, considerata la problematica attuale, lo scopo di questa tesi è quello di progettare e realizzare un sistema basato su ontologie al fine di (1) permettere l'interoperabilità tra sistemi eterogenei di dispositivi IoT per il fitness e il wellness, (2) facilitare l'integrazione e la condivisione delle informazioni e (3) consentire l'analisi avanzata dei dati raccolti (Cognitive Computing). Il contributo innovativo della soluzione proposta risiede nell'utilizzo delle tecnologie del Web Semantico per la descrizione formale del significato dei dati acquisiti mediante i dispositivi IoT e la definizione di una strategia condivisa di comunicazione per la rappresentazione e lo scambio dei dati.

Acknowledgments

First and foremost I want to express my deepest gratitude to my supervisor Professor Antonella Carbonaro for introducing me to the fascinating field of Semantic Web. I am also extremely thankful for her continuous guidance, motivation and trust.

I am grateful to Filippo Piccinini, PhD, Research Fellow at IRST Cancer Research Hospital of Meldola (FC, Italy), for his support and appreciation shown in my thesis, and for the long insightful friendly conversations we had about the scientific research.

My sincere gratitude goes also to all the authors of research articles who inspired me with their works.

I would also like to take the chance to thank my fellow university mates, for the time spent together following the lectures and working on university projects. I gratefully acknowledge their company and collaboration.

Finally, a special thanks goes to my parents for their support, advice and encouragement through my academic career.

Contents

Introduction

0.1 Motivation and Contribution

Nowadays, applications and systems exploiting sensors and producing data, are more and more popular. According to a Cisco prediction, by 2020, more than 50 billions devices will be connected to the Internet [62].

The potential of IoT technologies and deployment has already been seen in a copious number of different application areas such as transports, energy management, environmental monitoring, building and home automation, safety, etc. Among them all, medical care and healthcare do represent the most attractive application areas for the IoT [141]. In particular, healthcare is one major application sector for IoT identified by numerous researchers since the early stage of IoT innovations [12][56].

Mobile device assisted healthcare and medical applications are believed to create the next big industry progress due to increasing usage of mobile technologies and mobile devices. IoT technology is becoming one of the most popular trends also for the fitness area. Modern technology has revolutionised the fitness industry by giving users the ability to monitor themselves and keep track of their fitness training.

Besides that people have recently become more and more interested in their own health and fitness status. A transformation is underway regarding how we can deal with our health; mobile devices make it possible to have continuous access to personal health information.

More and more sophisticated wearable devices, fitness trackers, smart watches, heart rate monitors, electronic scales, innovative sleep monitors and fitness apps (mobile applications) will appear in the near future. Wearable devices, such as Fitbit[1] and Apple Watch[2] can collect data on 24/7 basis and provide insights into our health and fitness programs.

With billions of connected devices, thanks to *Cognitive Computing* techniques, IoT also promise to enhance decision making and data analysis to a level that was never achieved before. However, lack of interoperability and the presence of data silos prevent users, health professionals and researchers from getting an integrated view of health and fitness data.

To provide better health outcomes, it is essential to have a complete picture which combines informal health and fitness data collected by the user (Personal Health Record or PHR) together with official health records collected by health professionals (Electronics Health Record or EHR). To assist users or even machines in interpreting and combining these sensor data, there is a real need to explicitly describe sensor measurements according to the context, in a unified way in order to make them understandable to machines.

From a data-centric perspective the lack of worldwide acceptable standards keeps interoperability among IoT systems very limited. Data fusion and data integration of IoT silos is a burden to developers due to the fact that the IoT devices are highly heterogeneous in terms of data formats and data representation.

Despite the growing number of IoT deployments, the majority of IoT fitness applications and wellness devices tend to be self-contained, thereby forming application silos. In fact the main challenging problem is that devices are not interoperable with each other since their data is based on proprietary formats or they do not use common terms or vocabulary to describe the same concept.

The growing trend of *Linked Open Data* [116] also encourages to share the data on the Web, including medical data.

The aims of this thesis project is to design a lightweight ontology for the healthcare fitness domain in order to provide semantic interoperability among heterogeneous IoT devices, facilitate the integration and the sharing of data and enable advanced analytics over the collected data.

The novelty of this thesis lies in exploiting *Semantic Web* technologies [23] to deal with this challenge.

Semantic Web technologies have been chosen for several reasons: first

[1]https://www.fitbit.com

[2]https://www.apple.com/watch/

of all semantics enables an explicit description of the meaning of sensor data in a structured way, so that machines could understand it and more important semantic facilitates the interoperability among different devices and data integration since heterogeneous IoT data is converted according to the same vocabulary. Furthermore, Semantic Web technologies easily allow users to connect data, share information and knowledge on the web and making it machine-understandable using URIs, *RDF language* and *OWL ontologies*.

Moreover, the use of Semantic Web technologies has already been recognised as a successful approach for integrating health data among healthcare environments [134] such as the integration of IoT health and fitness data with the existing institution EHR systems.

0.2 Social Impact

When information technology meets healthcare and begins moving into the community, new opportunities arise to increase the global welfare.

In the last years, academia gave a revolutionary orientation to the University's courses, creating strong connections between computer and life sciences to form a new generation of workers and researchers with a solid knowledge on programming languages applied to face healthcare problems.

In fact, today, the number of computer science experts and engineers hired by hospital institutions for working to solve problems connected with data sharing and analysis is blooming.

It is well known that health services can be successful only if people are actually involved in their care and if the service providers recognise the diverse needs of individuals and their local communities to be efficient in offering what they really need.

Even more so on territories particularly receptive for health-related technology innovations such as the Emilia-Romagna (Italian region). As a matter of facts, Emilia-Romagna boasts one of the most advanced health network infrastructures in Italy, called SOLE[3]. SOLE interconnects all the local health authorities present on the territory both administrative and technical (i.e., labs operating within the regional health service, public medical

[3] `https://www.progetto-sole.it`

specialists, paediatricians and patients) and allows them to productively exchange information among each other.

The SOLE infrastructure has already gained the interest from academia and improvements to the system architecture and data model adopted have been proposed by Vitali et. al. in [178]. The current SOLE implements a document oriented approach, which means that the smallest addressable stored information is the document, the authors propose a shift to a data oriented approach that considers each single data directly addressable, while preserving the association with the original document, thus achieving a higher scalability and flexibility of the system.

However, the exploitation of the potential large amount of personal health and fitness data collected by IoT devices has still not been taken into consideration despite the numerous suggestions in the literature [154] [143] [58] [100].

PHRs enriched by IoT fitness devices data are person-centric tools that people can use to manage their own health status, thus becoming proactive participants in their own health management. The major benefits of PHR systems can be achieved when they are integrated with existing health institution EHR systems. Research has already widely demonstrated that integrated data can provide a more complete view of relevant health information for both consumers and their health care providers [108].

In this sense, the innovative system proposed in this thesis project, which combines the ubiquitous presence of the IoT fitness devices and the potential of the Semantic Web technologies, will have an undeniable potential beneficial impact on local communities and local healthcare institutions. This is to be expected, especially considering the fact that Emilia-Romagna has already successfully embraced the adoption of the *Fascicolo Sanitario Elettronico* (Italian equivalent for EHR) on a wide scale [55] [126].

Moreover, the Emilia-Romagna territory sees also the presence of many other numerous public and private health institutions which could benefit from the possibilities associated with IoT collected data for research purposes. Currently, the system designed in this thesis work is under extension in order to be then proposed for a testbed project involving oncological patients of the *Istituto Scientifico Romagnolo per lo Studio e la Cura dei Tumori* (IRST) of Meldola (FC, Italy), an Oncology Research Hospital strongly receptive for new systems, thus potentially improving the local healthcare quality.

0.3 Thesis Organisation

The thesis is organised as follows.

Chapter 1 presents a detailed overview of the current state of art of Semantic Web. The Semantic Web architecture is analysed by breaking it into its component parts and in particular RDF, RDFS and OWL languages are explained in depth. It draws attention to the concept of Open Data and its collocation in the context of the Semantic Web.

Chapter 2 introduces the main concept of ontologies and the role they play within the context of the Semantic Web. It overviews in details the Web Ontology Language along with its various dialects. It provides basic information about the reasoning engines. It briefly surveys the representative ontologies available in healthcare domain.

Chapter 3 gives an introduction to the Internet of Things technologies, in particular the role of the Internet of Things in the healthcare and fitness domain. Critical aspects of IoTs such as interoperability issues, from a data-centric perspective, are taken into a detailed consideration. Secondly it offers an overview of the most common IoT fitness devices available on the market.

Chapter 4 explains the concept of Semantic Data Annotation which is the key step for every Semantic Web project. It addresses the issues which arise when mapping systems have to deal with a plethora of heterogeneous data formats and briefly reviews the main features of two of the most common data serialisations within the IoT context. Finally, the RDF Mapping Language is proposed and analysed in depth.

Chapter 5 shows how all the notions given in the previous sections have been put together to build a framework system which aims to facilitate data integration and sharing, within the context of IoT fitness devices and wellness appliances. It gives an overview of the problem and it highlights the objective and the novelty of the proposed solution. After reviewing the previous works in the literature, it illustrates the architecture of the system and the details of each one of the principal components and the design process along with the motivation behind the choices made. Finally, it presents an overview of the advanced analytic techniques for IoTs within the research field of Cognitive Computing.

Chapter 6 summarises the main contributions of this thesis project and outlooks several possible directions for improvements and future works.

Chapter 1

Semantic Web

This chapter presents a detailed overview of the current state of art of Semantic Web. The Semantic Web architecture is analysed by breaking it into its component parts and in particular RDF, RDFS and OWL languages are explained in depth. Finally, this chapter draws attention to the concept of Open Data and its collocation in the context of the Semantic Web.

1.1 Semantic Web Overview

The World Wide Web (simply known as Web) has been developed back to 1990 by Tim Berners-Lee at CERN in Geneva, Switzerland.

The innovative idea behind the Berners-Lee's seminal work was to use *hypertext* [138] as a means to realise a distributed global system of interlinked documents accessible via the Internet.

On the Web documents are univocally identified by *Uniform Resource Locator* (abbreviated URL) addresses which specify how they can be retrieved across the Internet from their remote location. Documents are interconnected to each other by means of *hyperlinks* and URLs of the target resources are directly embedded in the body text.

The *HyperText Markup Language* (abbreviated HTML) is used to define the structure of the documents which primarily contain information in natural language, digital images, multimedia resources along with the rendering instructions to be displayed for human consumption.

Since its appearance on the Internet, the World Wide Web has become more and more mainstream and has grown into the world's largest repository of human knowledge. The rapid growth of the amount of information on World Wide Web has raised many research challenges such as information overloading, poor retrieval and aggregation problems. To find useful information is like trying *"to find a needle in a haystack"* for humans, due to the huge amount of data available and a hard task even for search engines which rely mostly on content-independent statistical algorithms. Syntactic variations or misspellings of the search keywords in documents prevent a reliable statistical score of document relevance.

Furthermore, users are often interested to retrieve data in aggregated manner instead of single separated documents. For instance, a user might be interested to find a smartwatch with certain features at the lowest price on the market. Performing such a task requires to gather information form several companies web pages, integrating their content and a kind of reasoning about the data obtained.

These issues derive from the fact that the current Web is mainly designed for human consumption and not for an automated machine processing, that is web pages do not provide any semantic information about the content which could allow machines to determine what the page content means.

The *Semantic Web* is an emerging research area which aims to overcome the challenge of allowing humans and computers to cooperate in the same way humans cooperate with each other. Tim Berners-Lee, the Web's inventor, has coined the term *Semantic Web* and in [23] provides a concise definition of it: *"The Semantic Web is not a separate Web but an extension of the current one, in which information is given well-defined meaning, better enabling computers and people to work in cooperation"*.

Berners-Lee envisages the World Wide Web as a collaborative medium by which users can share information and services easily and aggregating data from different sources where documents and web pages are understandable and processable by machines.

1.2 The Semantic Web Architecture

The Semantic Web is an extension of the existing Web (a *"syntactic Web"*) in which semantic is added to the resources.

The *Semantic Web Architecture*, as shown in Figure 1.1, is based on a layered approach, and each layer provides a set of specific functionali-

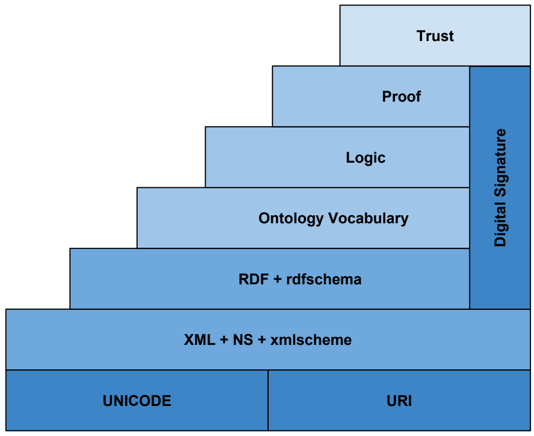

Figure 1.1: Semantic Web Architecture. (Image adapted from [157]).

ties. Several standards and technologies contribute to the realisation of the Semantic Web.

The lowest layers consist of data and metadata and provide a standard representation for information so that data can be easily exchanged among heterogeneous systems and applications.

The UNICODE provides a standard for a consistent encoding and representation of text expressed in most of the world's writing systems [42].

The URI provides a simple and extensible means for identifying and locating remote resources, such as web pages, media contents or other forms of data on the World Wide Web [125].

XML, the standard syntax for representing information in the Web, allows to structure data by means of user-defined tags and data interoperability [29].

The Resource Description Framework (abbreviated RDF) describes the information contained in a Web resource providing unambiguous methods to express semantics [131].

RDF Schema (abbreviated RDFS) allows to define simple vocabularies used in RDF descriptions [31].

Semantic layers, on the top of the stack, include ontology languages, rule languages, query languages, logic, reasoning mechanisms, and trust.

Ontologies constitute the backbone of the Semantic Web. Ontologies are a means to express concepts of a given domain and the relationships among the concepts and they also specify complex constraints on the types of resources and their properties.

OWL, the most popular ontology language, is an extension of RDFS. OWL Lite, OWL DL, and OWL Full are the three sublanguages of the OWL family ontology [129].

Rule languages allow writing inference rules in a standard way which can be used for reasoning in a particular domain. Among several standards of rule languages there are RuleML and SWRL (Semantic Web Rule Language) [91]. The latter combines RuleML and OWL, and includes a high-level abstract syntax for Horn-like rules.

SPARQL, a standardised query language for RDF data, provides both a protocol and a language for querying RDF graphs via pattern matching [149].

On the highest layers there are logic and reasoning, logic provides the theoretical underpinning required for reasoning and deduction. First order logic and description logic are frequently used to support the reasoning system which can make inferences and extract new insights based on the resource content rely on one or more ontologies.

Trust, Security, are needed to assure that the information content of resources is of high quality and can be trusted. More research is still to be done in order to develop comprehensive solutions and techniques to assess and ensure the trustworthiness, security, and privacy of Semantic Web content.

1.3 Resource Description Framework

The *Resource Description Framework* (abbreviated RDF) is a language for describing metadata about the resources in the World Wide Web and a W3C recommendation [123].

In a broader way, given that a resource is anything that can be referenced by a URI (Uniform Resource Identifier) [22], RDF is suitable to describe a resource of any type even when the resource can not be directly accessed from the Web [123].

RDF is mainly intended to be used when data need to be machine processable rather than being only accessed by people. Furthermore, RDF

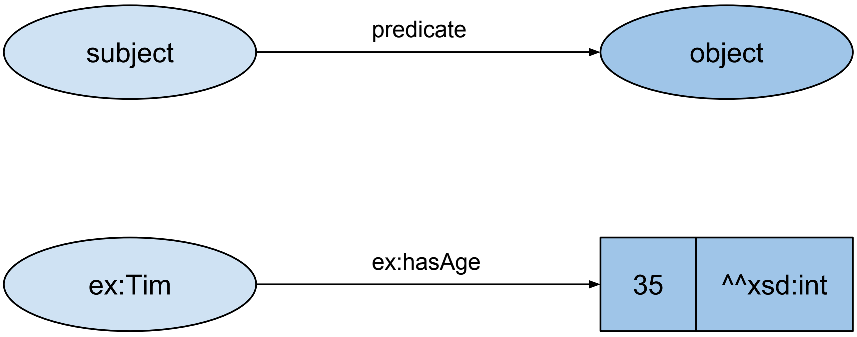

Figure 1.2: RDF graph of a generic triple and an example.

provides standardised way to express information such that it can be exchanged between different systems without loss of meaning [123].

RDF describes resources by means of triples. RDF triples have the form `(subject, predicate, object)` and provide the way to make *statements* about things.

Statements define the properties of the resources. A property expresses a relationship between the subject and the object. A property can designate a class to a resource, define a literal value attribute of a resource and a relationship between two resources.

The following example shows an RDF triple:

```
ex:Tim ex:hasAge  "35"^^xsd:int .
```

An RDF graph of the example above is depicted in Figure 1.2 along with a generic RDF triple.

Resources can be named or unnamed, the latter are represented with *blank nodes*.

```
_:bnode1 rdfs:label "anonymous" .
```

In the example above, `_:bnode1` denotes an anonymous resource, the prefix _ is used in many different RDF serialisation syntaxes to specify a blank node.

Naming and consistency are a significant part of RDF, user-defined resources are named using URIs and RDF supports CURIE syntax, which is an abbreviated syntax for expressing URIs [24]. For instance, given a prefix `ex:`, which acts as a shortcut for the URI `http://example.com/ontology#`, then `ex:heartRate` can be used instead of `http://example.com/ontology#heartRate`.

Datatypes in RDF are inherited from the existing XML Schema standard which defines a hierarchy of datatypes along with their syntax [25].

Language tagged strings in RDF should be defined in accordance with RFC 3066 [9] as shown in the following example:

```
ex:Italy
   rdfs:label "Italy"@en ;
   rdfs:label "Italia"@it .
```

RDF defines a core set of terms for describing resources, one of the most relevant is `rdf:type` which is used to state that a resource is a member of a specified class.

```
ex:Tim rdf:type foaf:Person .
```

The statement above asserts that the resource `Tim` is a member of the class `foaf:Person`.

RDF allows also to define containers which are used to describe groups (ordered, unordered or alternatives) of things with informally defined semantics [123], however since they are not widely used in practice they have been suggested as candidates for deprecation [64].

RDF collections are used to describe group that contains only the specified members. Unlike containers, collections may be closed and this is an important characteristic for reasoning.

Reification in RDF (describing RDF statements using RDF itself) is possible using the built-in terms: `rdf:Statement`, `rdf:subject`, `rdf:predicate` and `rdf:object`.

Applications may need to describe RDF statements, for instance, to record information like when statements were made, or who made them; other use cases where reification is useful are discussed in [120].

```
ex:TimAgeTriple rdf:type rdf:Statement .
ex:TimAgeTriple rdf:subject ex:Tim .
ex:TimAgeTriple rdf:predicate ex:hasAge .
ex:TimAgeTriple rdf:object "35"^^xsd:int .
ex:TimAgeTriple ex:expires "2017-12-15"^^xsd:date .
```

The example above shows the reification of the statement:

```
ex:Tim ex:hasAge "35"^^xsd:int .
```

Even though properties in RDF are only binary relations (relations between two classes), *n-ary relations*, to link an individual to more than just one individual or value are possible by creating an intermediate entity that serves as the subject for the entire set of relations [139].

The following example introduces a blank node to model a tertiary relationship:

```
ex:Tim ex:hasHeight _:bnode1 .
_:bnode1 rdf:value ex:Measure .
_:bnode1 ex:numericalValue "186"^^xsd:int .
_:bnode1 ex:Unit ex:cmUnit .
```

RDF triples can also be put together to form larger networks also known as *semantic networks*. A semantic network is a direct graph where vertex are the subject or the object of a triple and edges are labelled with predicates and are directed from the subject to the object.

RDF is an abstract model and RDF statements can be represented either as a graph or in a textual format also called RDF serialisations. The most important RDF encoding syntax is RDF/XML [19] which is based on Extensible Markup Language (abbreviated XML) standard [29] and currently is the only normative RDF encoding standard. Other notable RDF serialisations syntax are: N-Triples [33] which is a line-based (a single statement cannot span multiple lines) plain text serialisation, Turtle (see Section 1.3.1) and RDFa which allow to embed RDF statement within an XHTML document [4].

1.3.1 Terse RDF Triple Language

A notable RDF textual format representation, besides the most common serialisation RDF/XML, is *Turtle* (Terse RDF Triple Language) [18].

Turtle defines a syntax which allows a completely compact textual representation of RDF graphs, in both machine and human readable format.

Turtle syntax has also been used extensively throughout this thesis.

The salient characteristics of the Turtle syntax are briefly review:

- URIs are written surrounded by `< >` brackets.

  ```
  <http://example.com/fitnessOntology#Walking>
  ```

  This statement represents a walking activity entity.

- Namespaces can be declared to prefix URI using `@prefix`

  ```
  @prefix fo: <http://example.com/fitnessOntology#> .
  ```

- Tokens and terms are white-space delimited and triples are delimited by a `.` period character.
- Literals are represented between `" "` double-quotes.
- Literals can be typed by XSD datatypes; assigned datatypes are appended after a `^^` operator.
- The _ underscore prefix is used to denote blank nodes.

  ```
  _:bnode1 rdfs:comment "anonymous"^^xsd:string .
  ```

- The term `a` can be used as a shortcut for `rdf:type`.

  ```
  ex:Tim a foaf:Person .
  ```

- Triples which share a common subject and predicate can be grouped together using a `,` comma delimiter.

- Square brackets `[]` can alternately be used to denote blank nodes.

```
[ rdfs:comment "blank node"^^xsd:string,
               "another comment"^^xsd:string ] .
```

- Triples which share a given subject can be grouped together using a ; semi-colon delimiter.

```
ex:Tim a foaf:Person;
         rdfs:comment "someone"^^xsd:string .
```

1.4 Resource Description Framework Schema

The *Resource Description Framework Schema* (abbreviated RDFS or RDF Schema) [31] is a language for defining simple *vocabularies* (which are a kind of ontology) of terms that can be use to construct RDF statements according to these ontologies.

RDF Schema is an extension of RDF, it is expressed in RDF syntax, and provides the means for specifying well defined relationships between classes and properties in a hierarchical structure.

RDFS allows users to define classes and properties (predicates) using the relations `rdfs:Class` and `rdfs:Property`. A *class* is a set of things, sharing common characteristics, that we want to represents; a property is a binary relation between two class individuals. Individuals are instances of a class, which means that they are objects that belong to a particular class, are defined by assigning the type of a class to the resource through `rdf:type`.

In RDFS a class `C` is defined by a triple of the form:

```
C rdf:type rdfs:Class .
```

For example a class to represent "users" can be as follows:

```
ex:User rdf:type rdfs:Class .
```

A class in RDFS represents a set of resources and the hierarchy defines the relationship between different classes. RDFS is structured around the notion of a class hierarchy. A *subclass* is a class that has to be intended as a subset of the more general class and is specified by the property `rdfs:subClassOf`. The subclass relation is also the only relationship between classes that RDFS allows.

```
ex:User rdfs:subClassOf ex:Person .
```

This states that any member (also called *instance*) of the class `ex:User` is also a member of the class `ex:Person`.

In a similar way, RDFS allows the definition of a hierarchical structure also for properties in addition to the hierarchy of classes. That is, using the relation `rdfs:subPropertyOf` we can state that a property is more specialised than another.

```
ex:directorOf, rdfs:subPropertyOf, ex:worksFor .
```

This triple states that two objects related by the `ex:directorOf` property are also related by the `ex:worksFor` property.

Furthermore, RDFS allows to put restrictions on the properties to a certain classes of resource using the relations `rdfs:domain` and `rdfs:range`; which means that the domain and range of the property is restricted to specific classes.

Other properties introduced to make RDFS document more human-readable are: `rdfs:comment` which allows to give an informal description of the resource, `rdfs:label` for specifying an alternative labelling scheme, `rdfs:seeAlso` to reference another resource which provides related information and `rdfs:isDefinedBy` which is also a subproperty of the former and used to indicate that the definition of the resource is given elsewhere (e.g., in a book).

It is worth to mention that RDFS schema definitions are not *prescriptive* [144]. The RDFS schema is a merely description of the structure of the knowledge and it is let to the external application to decide whether to insist on full compliance with the schema or not. Because of the flexible nature of Semantic Web knowledge, it is perfectly acceptable to structure the knowledge base adding classes or properties outwith the schema or even violate specific constrains.

1.5 Linked Open Data

The idea behind the *Open Data* (abbreviated OD) is closely similar to the concept of the open source software [34]. According to the Open Definition the essence of open data can be summed up in the statement: *"Open means anyone can freely access, use, modify, and share for any purpose (subject, at most, to requirements that preserve provenance and openness)"* [2].

Jansenn et al. define Open Data as *"non-privacy restricted and non-confidential data which is produced with public money and is made available without any restrictions on its usage or distribution. Data can be provided by public and private organisations, as the essence is that the data is funded by public money"* [98].

Open Data refers to publish any collection of data in a machine-readable format, with no licensing or patent restriction so that everyone is free to use, reuse and redistribute for any purpose.

Governmental organisations, individuals, companies and enterprises are continuously gaining interest in Open Data recently. Governments provide transparency and increase increase public participation through Open Data. Scientific institutions can benefit from Open Data for deriving new knowledge and insights. Entrepreneurs can use the data to support their business, strategic decisions and foster innovations.

An exhaustive survey about Open Data benefits and the challenges in adoption of it can be found in [98].

Strictly related to the concept of Open Data is the concept of *Linked Data* (abbreviated LD). Linked Data refers to *"data published on the Web in such a way that is machine-readable, its meaning is explicitly defined, it is linked to other external datasets, and it can in turn be linked to from external datasets"* [26].

The merger of the movement of Open Data with the concept of Linked Data gives raise to a powerful data organisation and knowledge distribution. The *Linked Open Data* (abbreviated LOD) as the combination of Open Data and Linked Data is a method of publishing machine-readable open data so that it can be interlinked among different datasets on the Web enabling data integration and semantic querying [26].

In the context of the Semantic Web, data should be available in Resource Description Framework (RDF triples) which also provides the possibility of querying the datasets using SPARQL. Data are also univocally identified

Table 1.1: 5-star Open Data

Stars	Data Characteristics
★	open license, any format
★★	structured format
★★★	non-proprietary open format
★★★★	URIs to identify resources
★★★★★	data interlinked to provide context

by means of URIs and transferred through the HTTP protocol.

In 2010 Tim Berners-Lee proposed the *five-stars model* [21] which classifies Open Data into five different categories depending on the format on which data is distributed and is now widely accepted as framework evaluate quality of LOD projects. The five-stars classification schema is summarised in Table 1.1.

Chapter 2

Ontologies

This chapter introduces the main concept of ontologies and the role they play within the context of the Semantic Web. It overviews in details the Web Ontology Language (OWL) along with its various dialects. It provides basic information about the reasoning engines. Finally, it briefly surveys the representative ontologies available in healthcare domain.

2.1 Overview

The word *Ontology* comes from the Greek *ontos* (being) and *logos* (study) and has its root in philosophy where it refers to the subject of being and existence as well as the basic categories [183]. In other words, the term Ontology is used to refer to *"the study of categories of things that exist or may exist in some domain"* [165].

Even though there is no universal definition for ontology, one of the most frequently cited in the Semantic Web literature is the one proposed by Gruber et al.: *"an ontology is a formal, explicit specification of a shared conceptualisation"* [79]. Here, *conceptualisation* stands for a simplified representation or an abstract model of the world within the domain considered; *shared* because it has to captures consensual knowledge (i.e., it is accepted by a group and not only by a single individual). Ontology is also an *explicit specification* which means that objects, concepts and relationships must be clearly defined; and *formal* indicates that the ontology should be machine understandable.

Ontology is also a well-known concept in artificial intelligence and in particular in the knowledge representation field. Knowledge engineers intend with ontology a means for representing knowledge in a way that machines can reason, that is, making inferences and valid deductions.

Uschold et al. highlight that an ontology, despite the several different formats it may assume, normally include a vocabulary of terms, specifying their meaning and indicating how they are interrelated [101]. More simply, an ontology is the representation of the knowledge according to a specific domain, where the concepts and their relationships are described by a vocabulary.

Within the context of the Semantic Web, ontologies categorise concepts into classes based on common attributes and characteristics reflecting the George Lakoff's "classical vision" of categorisation [30]. According to Lakoff's vision, a class is defined by a set of properties and the basic condition for an object to belong to a class is to possess all the properties associated with the class [114]. Properties may be defined as necessary and sufficient so that inference mechanisms will automatically identify membership.

Semantic Web ontologies enable machines to interpret and process information on the Web, providing a common model that can be understood both by humans and computer, to share, exchange, and reuse data based on their intended meanings.

The use of ontologies aims at achieving *semantic interoperability* by bridging and integrating multiple and heterogeneous digital content on a semantic level, which is exactly the core idea of the Semantic Web vision. Furthermore, not only the use of ontologies reduces the semantic ambiguities by offering a single interpretation resource, but also, information content is made available for machine consumption, whereas the majority of the content found on the Web today is primarily intended for human consumption only.

2.1.1 Taxonomies and Thesauri

This section discusses the distinction between the concepts of taxonomy and thesaurus. Even though taxonomies and thesauri are not specifically designed for the Web, in fact they don't appear on the Semantic Web stack, they, however belong to the Semantic Web picture.

Taxonomies

Daconta et al. define *taxonomy* as: *"the classification of information entities in the form of a hierarchy, according to the presumed relationships of the real-world entities that they represent"* [49]. A taxonomy provides a means to categorise, organise, label, and arrange information in hierarchical fashion using *father-son* relationships. A father-son relationship is a generalisation for the *is-a* and the *type-of* relationships, and is the only one kind of relationship which hold among concepts ruling out other relationships, such as *part-of*, cause-effect, association, and localisation. Furthermore, taxonomies do not permit defining attributes for terms.

Below an example of taxonomy; the classification of the human species in the Linnaean living being taxonomy[1]:

```
Kingdom: Animalia
        Filo: Cordata
            Subfilo: Verebrata
                Class: Mammalia
                    Subclass: Theria
                        Order: Primata
                            Suborder: Anthropoidea
                                Family: Hominidae
                                    Genera: Homo
                                        Species: Sapiens
```

Note that all the terms present are related by the generalisation relationship (e.g., *Mammalia* is a type of *Vertebrata*, which in turn is a type of *Chordata*, which in turn is a type of *Animalia*).

Thesauri

According to the ANSI/NISO Monolingual Thesaurus Standard a *thesaurus* is defined as: *"a controlled vocabulary arranged in a known order and structured so that equivalence, homographic, hierarchical, and associative relationships among terms are displayed clearly and identified by standardised relationship indicators ..."*.

[1]https://en.wikipedia.org/wiki/Linnaean_taxonomy

In other words, a thesaurus can be seen as a taxonomy together with a set of semantic relationships, such as equivalence, inverse, and association, that hold among the concepts.

A thesaurus can be used to guarantee that concepts are described consistently to enable users to refine searches and locate the information they need [30].

If relationships other than those thesauri support (i.e., equivalence, homographic, hierarchical, and associative relationships) are required, one must resort to more general ontologies.

A notable example of a thesaurus is WordNet[2]. WordNet is a thesaurus for the English language based on psycholinguistics principles and developed at the Princeton University by George Miller [133]. WordNet is an online lexical database designed for use under program control. English nouns, verbs, adjectives, and adverbs are organised into sets of synonyms, each representing a lexicalised concept. Semantic relations link the synonym sets [132].

2.1.2 Ontologies Classification

In the literature, various different ontology classifications exist. As depicted in Figure 2.1, Guarino et al. propose a classification based on the degree of generalisation [80]:

- *Top Level Ontologies*: describe very generic and abstract concepts such as space, time, matter, object, event, action, etc. Ontologies of this kind are valid regardless of the specific problem or domain of interest.

- *Domain Ontologies*: describe a vocabulary related to a generic domain (e.g., medicine or a sport) by specialising the concepts provided by the top level ontology.

- *Task Ontologies*: describe the vocabulary of terms needed to perform generic tasks or activities (e.g., diagnosis) by specialising the concepts provided by the top level ontology.

[2] WordNet is a registered trademark of Princeton University.

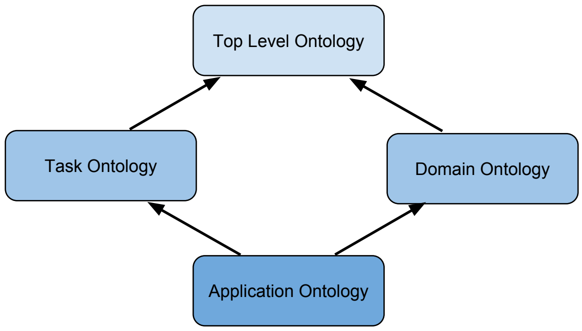

Figure 2.1: Guarino's ontology classification. Thick arrows represent specialisation relationships. (Image adapted from [80]).

- *Application Ontologies*: describe the terms of concepts depending both on a particular domain and task. Ontologies of this kind are restricted only to a specific application.

McGuinness et al. propose a classification based on the internal structure of the ontologies; ontologies range from lightweight to heavyweight, depending on the complexity which characterises the elements they contain [127]. According to Corcho et al. a lightweight ontology is composed by concepts, properties, relationships and concepts taxonomies, while heavyweight ontologies are complex and include also axioms and constraints [44].

Gomez et al. suggest a classification which is partially orthogonal to the previous discussed above and it is based on the information represented by the ontology [77].

It is noteworthy to highlight that clear lines among these categories cannot be drawn, neither is there any formal specification to classify ontologies.

2.2 Reasoning

Reasoning is the process of extracting new knowledge (inferring facts that have not been explicitly stated) from an ontology and its instance base and is one of the most powerful features of Semantic Web technologies.

A *Semantic Reasoner* (also known as *reasoner engine* or simply *reasoner*) is a software system whose primary goal is to infer knowledge which is *implicitly* stated by reasoning upon the knowledge *explicitly* stated, according to the rules that have been defined.

The reasoners are also used to *validate* the ontology, that is, they check its consistency, satisfiability and classification of its concepts to make sure that the ontology does not contain any inconsistencies among its term definitions.

According to Donini et al. the basic ontology reasoning procedures [57] can be listed as follows:

- *Consistency checking*: assures that the ontology does not contain contradictory facts (e.g., equality and inequality assertions).
- *Concept satisfiablility*: checks whether a class can have at least one individual or not. Having unsatisfiable classes usually means that the entire ontology is not consistent.
- *Concept subsumption* (*classification*): determines the subclass relationships between classes in an ontology in order to complete the class hierarchy.
- *Instance checking*: checks whether an individual is an instance of a class (i.e., it calculates the individual type).
- *Conjunctive Query Answering*: answers a (SPARQL) query with regard to an ontology.

As far as Description Logics (and Logics in general) are concerned, desirable properties of these reasoning techniques are:

- *Termination*: is related to guarantee that for a given input the algorithm can terminate.
- *Soundness*: ensures that every formula proved to be satisfiable, is indeed satisfiable.
- *Completeness*: concerns to the capability of deducing every possible fact that can be inferred from the available set of axioms.

A lot of research is currently being focused on investigating the compromise between the expressiveness of ontology definition languages and the computational complexity of the reasoning procedure, as well as the discovery of efficient reasoning algorithms applicable to practical situations [111].

Three classical open source reasoners available are: HermiT [158], Pellet [162] and FaCT++ [173].

2.3 Ontology Web Language

RDFS is deliberately intended to be a simple language to define ontologies such as vocabularies and taxonomies but in many cases to address the demands of the Semantic Web more expressiveness is needed. The *Web Ontology Language* (abbreviated OWL) [129] is an ontology language which extends RDFS to overcome its limitations. OWL is a W3C Recommendation and is the *de facto* standard for publishing and sharing ontologies in the Semantic Web.

OWL as a markup language for specifications of ontologies has been used for applications in a large variety of fields such as medicine [75], biology [161], agriculture [164] and defence [113].

OWL mainly derives from DAML+OIL Web Ontology Language [128] [92] which in turn is a combination of DAML [86] and OIL [65].

Like RDF Schema, OWL can be serialised using RDF syntax and adopts the *open world assumption* (which means that missing information is treated as unknown) and the *not unique name assumption* (different identifiers may refer to same entities in the real world).

OWL introduces many new language primitives which extend RDF and RDFS. OWL allows to define classes as a combination of other classes using set operators like union, intersection and complement. In OWL is possible to state that two classes are disjoint or are the same (despite being identified with different URIs). It is also possible to use restrictions on properties such as *cardinality* or specify that a certain property is *transitive* or *unique*.

The Web Ontology Language provides richer schema for expressing meaning and semantics but the more expressive is a language, the more is difficult to reason with the language. Although complex language constructs allows to represent more knowledge, computation becomes inefficient and eventually undecidable.

When it comes to choosing and ontology language for the Semantic Web there is always a trade-off between expressibility and efficient reasoning, depending on the kind of application to be designed.

OWL consists of a family of three languages with different degrees of expressivity and computational properties: OWL Full, OWL DL and OWL Lite.

OWL Full

OWL Full is the most expressive language it places no restrictions on how the language constructors can be used. This flexibility comes at the expense of decidability, in fact reasoning task such as consistency checking, satisfiability checking, subsumption checking, instance checking and conjunctive query answering. All of these typical reasoning tasks over an OWL Full ontology are undecidable.

OWL DL and OWL Lite are two restricted forms of the OWL language, restrictions make them "decidable". Both OWL DL and OWL Lite are based on *Description Logic* (abbreviated DL) [13] which guarantees (all conclusions are guaranteed to be computable) and decidability (computation will be finished in finite time).

OWL DL

OWL DL is the more expressive after OWL Full and is also the most important among the three variants of the OWL family. OWL DL is equivalent to a well-defined DL and contains all of the OWL language primitives but allows restricted use of them. A full list of restrictions put in OWL DL can be found in [129].

OWL DL is decidable for consistency, satisfiability and instance checking tasks. However the complexity of these reasoning task are NExpTime-complete which means that for certain valid inputs the reasoning task may not be completed in "acceptable time".

OWL Lite

OWL Lite is a subset of OWL DL and it is most restricted variant of OWL. The rationale behind OWL Lite is to trade expressivity for efficiency of reasoning: *"reasoners for OWL Lite will have desirable computational properties"* [181]. The complexity of OWL Lite is ExpTime-complete for consistency, satisfiability and instance checking tasks. As opposed to OWL DL, conjunctive query answering is decidable, however OWL Lite 2ExpTime-complete with respect to query complexity [47] which means that for certain valid inputs, despite the certainty of decidability, reasoning is intractable.

2.3.1 OWL 2

OWL 2 [89], addresses in part the issues which afflict the previous version of the language and introduces new language primitives and semantics for OWL 2 Full and OWL 2 DL. A comprehensive report of the rationale and new features introduced bye OWL 2 can be found in [74], here is given a brief overview.

While OWL 1 defines only two main dialects OWL Full and OWL DL one syntactic subset (OWL Lite), OWL 2 provides in addition three new *profiles*: OWL 2 EL, OWL 2 QL, and OWL 2 RL. These profiles are syntactic subsets of OWL 2 DL and are intended to target different application scenarios by trading the expressivity to achieve an efficient reasoning.

OWL 2 EL

OWL 2 EL is based on the Direct Semantics [136] and it is primarily designed for dealing with a large number of class axioms and classification tasks (such as subsumption and instance checking).

OWL 2 EL was conceived to address the complexity of numerous existing large-scale ontologies in the healthcare and life sciences domain such as SNOMED-CT[3] [163] or Gene Ontology[4] [41].

Reasoning for OWL 2 EL is PTime-complete (polynomial complexity) except for query-answering [136].

OWL 2 QL

OWL 2 QL is also based on the Direct Semantics and provides more expressive features such as the property inclusion axioms and functional and

[3]SNOMED-CT is an ontology of clinical terms with over 500000 classes.

[4]Gene Ontology is a biological ontology that describes genes and gene properties with more than 25000 classes.

inverse-functional object properties.

The QL profile of OWL2 was developed to efficiently handle query answering in ontologies which contain a large number of individual assertions and relatively uncomplicated class definitions. OWL 2 QL also adopts technologies from relational database management.

Reasoning is NLogSpace-complete with the exception of query answering which is NP-complete [136].

OWL 2 RL

OWL 2 RL is based on *Description Logic Programs* (abbreviated DLP) as proposed by Grosof et al. [78] and pD* proposed by ter Horst et al. [170].

OWL 2 RL enables interaction between description logics and rules, in fact it was primarily designed to deal with ontologies that describe rules within. OWL 2 RL is basically a rule language and rules can efficiently be run in parallel, allowing for scalable reasoning implementations.

Reasoning in OWL 2 RL is PTime-complete except for query answering which is NP-complete [136].

2.4 Healthcare Ontologies

Due to the extreme complexity of medical terminology systems and medical information systems, ontologies play a central role for the representation, management, and sharing of knowledge and data.

Ontologies are preferred to conventional classifications due to the higher level of expressiveness that is possible to achieve in describing concepts and their relationships. Furthermore, the domain knowledge in a machine processable format facilitates an efficient information retrieval.

In the past years, a plethora of healthcare domain ontologies have been created. Such representations are used to systemically denote, categorise, and relate healthcare data, allowing easier handling of the data in healthcare information systems [52].

Most of the existing healthcare ontologies are designed to describe a specific domain in biomedicine, such as the terms to describe anatomical parts and their relations, or terms used in clinical medicine, such as in EHR (Electronic Health Records) systems or rehabilitation domain [187].

Healthcare ontologies are widely recognised as a key factor technology to provide the semantics required for deriving proper treatment through integrating clinical guidelines [95].

The number of ontologies in the healthcare domain is constantly increasing; BioPortal provides access to a library of biomedical ontologies and terminologies developed in Web Ontology Language (OWL), Resource Description Framework Schema (RDFS), Open Biological and Biomedical Ontologies (OBO) format [182].

Below the main characteristics of SNOMED-CT and LOINC ontologies are briefly reviewed.

SNOMED-CT

The *Systematized Nomenclature of Medicine-Clinical Term*[5] (abbreviated SNOMED-CT) is considered as the main ontology for a standardised representation and automatic interpretation of clinical concepts, terms and relationships in the field of health care.

The ontology covers most of the areas that are used in medical practice, including clinical findings, symptoms, diagnoses, pharmaceuticals, body structures, medical devices, social contexts, and so on.

SNOMED-CT has hierarchy structure with a set of top level general concepts. All other concepts are subtypes of one these top concepts. Each concept is assigned a unique ConceptID and a Fully Specified Name (FSD).

SNOMED-CT provides a consistent way for indexing, storing, retrieving and aggregating clinical data that can enhance the interoperability between different health information systems.

LOINC

The *Logical Observation Identifiers Names and Codes*[6] (abbreviated LOINC) is a universal code system for laboratory test and other clinical observations. For each observation provides a code, a short name, a long formal name and synonyms.

The primary purpose of LOINC is to provide common codes and terminology which allow hospitals, pharmaceutical manufacturers, researchers,

[5] `http://www.ihtsdo.org/snomed-ct`
[6] `https://loinc.org`

and public health departments to receive clinical observations from multiple sources, so that they can automatically file the data in the right slots of their medical records, research, and public health systems.

Chapter 3

Internet of Things

This chapter gives an introduction to the Internet of Things technologies, in particular the role of the Internet of Things in the healthcare and fitness domain. Critical aspects of IoTs such as interoperability issues, from a data-centric perspective, are taken into a detailed consideration. Secondly it offers an overview of the most common IoT fitness devices available on the market.

3.1 IoT

The term *Internet of Things* (abbreviated IoT), sometimes also referred to as *Internet of Objects* or *Smart Objects*, denotes **any combination of software and hardware that produces data through connecting multiple devices and sensors**.

The term Internet of Things was first introduced by Kevin Ashton at the Auto-Id centre of the Massachusetts Institute of Technology (MIT) back in 1999.

Anything can be an IoT device if it can transmit and receive data over the Cloud or, in other words, any system that can connect objects or things to Internet, hence connecting the physical world to the virtual world.

Internet as a medium to communicate and exchange information is a living entity, constantly changing and evolving, and now is shifting from only connecting people and computers towards connecting *things* and *objects*.

This vision where objects become a part of the Internet is also possible due to an unceasingly evolving technology: Internet broadband connectivity

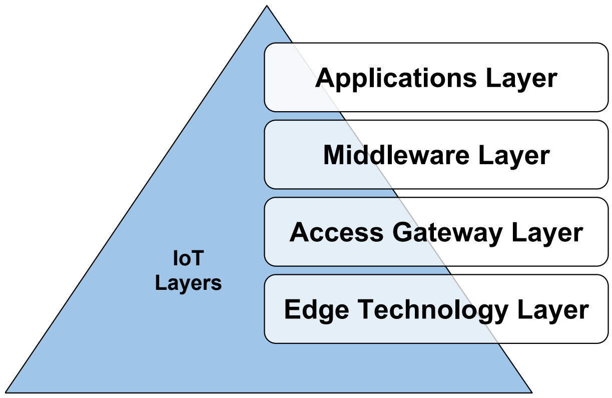

Figure 3.1: Layered Architecture of an IoT System. (Image adapted from [157]).

is becoming cheaper and ubiquitous, devices are becoming smaller and more energy efficient and fitted with a large variety of on-board sensors.

IoT is nowadays recognised as one of the technologies that will radically and permanently transform our life, business, and the global economy in the near future [124].

IoT paradigm can be applied to a long list of different domains ranging from transportation, supply chain, environmental monitoring, inventory and production management, smart cities, smart homes, building automation, data collection, social networks to medical care, healthcare. These latter ones in particular represent one of the most attractive application areas for IoT [141]. More and more of new applications and businesses for IoT are created continuously.

3.1.1 IoT Architecture

As shown in Figure 3.1, Santucci et al. describe the architecture of a typical IoT device as a four layered architecture: the *edge technology layer*, the *access gateway layer*, the *middleware layer* and the *application layer* [157].

The two lowest layer are responsible for data collecting and network connectivity while the two highest layers are responsible for data utilisation in applications.

The functions of every single layer (from the lowest to the highest) are as follows:

1. *Edge Technology Layer*: this layer is also known as *perception layer* is the hardware layer which includes components for network connectivity, data storage and data collection through sensors such as GPS, cameras, pressure sensors, temperature sensors etc. The Edge Technology Layer also provides information processing (via embedded edge processor), control and actuation.

2. *Access Gateway Layer*: this layers is also known as *network layer* or *transport layer* and is responsible for data transmission and routing. It receives information from the edge layer using communication technologies such as Wi-Fi, Li-Fi, Ethernet, GSM, WSN, ZigBee Bluetooth and WiMax [157][12] and sends them do the middleware.

3. *Middleware Layer*: this layer provides an abstraction to applications from things. It also provides services such as data filtering, data aggregation, semantic analysis and access control.

4. *Application Layer*: which is also the top layer of the stack consists of two sub-layers: the *data management sub-layer* and the *application service sub-layer*. The data manager sub-layer provides directory service, quality of service (QoS), cloud computing technologies, data processing, machine-to-machine (M2M) services etc. The application service sub-layer on the other hand is responsible for interfacing the system to end users and enterprise applications running on top of the IoT applications layer.

Cloud and Fog Computing for IoT

Information processing is handled in application layer. The information processing technologies for IoT applications include also Cloud Computing and Fog Computing. For instance, in the case of healthcare applications that depend on utilisation of inputs from the physical world (e.g., vital signs of a patient via sensors), a huge amount of data is constantly collected. The data can be sent to a cloud integrated with the IoT system for a safe, convenient and efficient storage, processing and management [59]. The cloud-based approach enhances healthcare solutions by improving accessibility and quality of healthcare, and reducing costs [71].

Similarly, fog computing extends cloud computing. It is a distributed computing infrastructure that provides the same application services to end-users as cloud computing such as data processing, storage, and execution of applications. However, the application services are handled at the network edge in a smart device instead of a remote datacenter in the Cloud. The goal of fog computing is to improve the efficiency and reduce the amount of transported data to the Cloud [27].

3.1.2 IoT Critical Issues

Despite the growing number of IoT devices and applications, IoT technology is still in its infant stage and has big room for research in variety of issues such as standards, scalability, heterogeneity of different devices, common service description language, safety and integration with existing IT systems just to cite a few.

Interoperability as the ability to interconnect and communicate different vendors' system along with data integration is one vital issue still unsettled.

Barnaghi et al. highlight four interoperability issues in IoT [16]:

1. *Technical interoperability* involves the heterogeneity of hardware and software components and the related communication protocols.

2. *Syntactical interoperability* involves data formats and data representation. Syntactical interoperability is crucial to interpret IoT data and build smart systems. They underline the need to agree on common vocabularies to describe data.

3. *Semantic interoperability* involves the interpretation of meaning of data exchanged.

4. *Organisational interoperability* involves the heterogeneity of the different infrastructures. Organisational interoperability depends on successful technical, syntactical and semantic interoperability.

In this thesis, we only address the syntactical and semantic challenges. Nowadays the majority of IoT applications tend to be self-contained thereby forming application silos [177]. Chen et al. state : *"cannot correlate and integrate the data from different silos and getting the data from heterogeneous* *sources"* [36]. They highlight the needs for IoT data processing and explain the issue related to domain specific-applications: applications cannot combine the data from different silos.

Sensor data are useless if we do not analyse and understand them correctly. Interpreting raw IoT data, extracted from devices, in a meaningful way is still an open issue and a challenge [69].

Interoperability can be solved if communicating smart systems are semantically interoperable [70]. Semantics gives a structure to data and captures the meaning.

This challenge is particular relevant in the health care and fitness domain where a multitude of diverse vendor devices collect the same type of data but store and exchange them in many different ways, so there will be semantic and syntactic conflicts.

Semantic Web technologies are promising tools for this purpose to share data and exchange their services efficiently [99]. Semantic Web technologies are also the approach that has been adopted for this thesis project.

3.2 Web of Things

The main IoT's goal is to connect physical devices to the Internet. The concept of *Web of Things* (abbreviated WoT) [186] concerns the connection of the sensors specifically to the web, getting the data and exchanging the data ,that has been produced by devices, on the web.

Existing web technologies can be adapted and reused to build new smart applications and services exploiting data generated by the IoT devices by integrating Smart Things to the Web. Web services have been proven to be crucial in creating interoperable applications on the Internet, IoT devices can be abstracted as web services and seamlessly integrated into the existing web. The WoT vision depicts a view where a collection of web services can be discovered, composed and executed.

There are two possible methods to integrate things into the Web: *direct integration* and *indirect integration* [81]. In many cases IoT systems use both methods has a hybrid way.

Direct integration means integrating things into the Web using embedded web servers. IoTs running an embedded web server can directly communicate with the users from any terminal with a standard web browser. Other devices can also inter-operate with them through standard web operations specified by web standards (e.g., GET and POST). Web severs can be built in a size of only a few KBs [60] [5] so that they can be easily embedded into many devices directly despite the limiting memory and computational capabilities of the IoT devices. Indirect integration on the other hand is needed when a device is not powerful enough to be embedded with a web server. Sometimes there is also no need to directly integrate all the smart things to the Web in the consideration of cost, energy and security[81]. For indirect integration, an *intermediate proxy* (also called smart gateway) placed between things and the Web is used. The proxy communicates directly with the smart things, this implies that it understands the proprietary protocols of the devices, and exposes outward to the Web the proprietary protocols and the native APIs of the smart things abstracting them. In this way IoT can still be accessible using web standards.

3.2.1 Semantic Web of Things

Semantic Web of Things (abbreviated SWoT) is a research field which aims to combine Semantic Web technologies to Internet of Things providing interoperability among ontologies and data [99][146].

Existing WoT systems deal with heterogeneous protocols and easily share sensor data on the Web. However, they do not use Semantic Web technologies. SWoT differs from WoT by adding semantics in order to ensure a common understanding. Semantic Web of Things can be seen as an evolution of Web of Things through integration of IoT with web technologies to access the devices and the produced data via Web.

Barnaghi et al. show that semantic is needed to: (1) provide unambiguous IoT data descriptions to be interpreted by machines, (2) combine data from different physical systems and devices, (3) semantic reasoning, and (4) sensor discovery [17].

SWoT promises a seamless extension to the IoT allowing integration of both the physical and digital worlds and are focused on providing wide scale interoperability that allows the sharing and reuse of data [99].

The SWoT vision enables also knowledge-based systems to achieve high degrees of autonomic capability for information storage, management and discovery leveraging on ontologies and standardised semantic web languages such as RDF, RDFS and OWL.

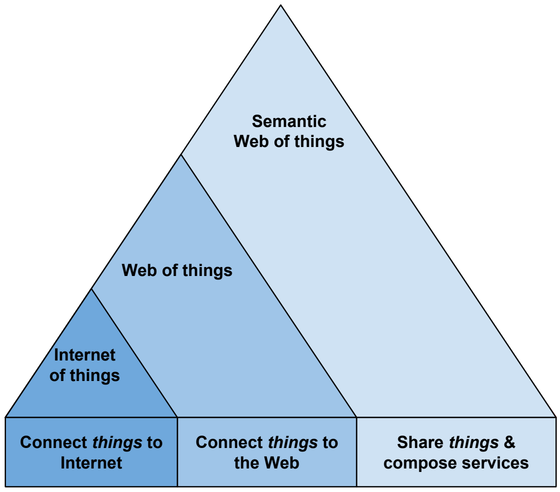

Figure 3.2: From IoT to SWoT. (Image adapted from [99]).

3.3 IoT and Healthcare

The healthcare industry has seen a radical change in the era of the Internet with relatively cost efficient and smart solutions such as IoT technologies.

According to Atzori et. al the healthcare domain has a huge potential for IoT successful applications and smart systems [12].

Koop et al. envision a new delivery model of healthcare, enabled by the IoT technology, that will transform the present hospital-centric, through hospital-home-balanced by 2020, to the final home-centric by 2030 [112]. Healthcare providers around the world are transforming themselves into more efficient, coordinated and user-centred systems and technology plays a central role in achieving efficiency and enhancing distributed healthcare smart systems that fulfil diverse and constantly increasing demands.

Significant segments of the IoT healthcare market are: independent living services, consumer medical devices, telemedicine, wearable technologies, fitness monitoring devices, health gaming, personal emergency response systems and wearable technologies [66].

Typical IoT solutions for healthcare can be categorised as followings:

- *Tracking and monitoring*: thanks to the ubiquitous identification, sensing, and communication technologies patients and people can be tracked and monitored by wearable devices on a 24/7/365 basis [8]. Wearable fitness tracking devices and life logging devices belong to this category.

- *Remote service*: healthcare and home assistance, emergency detection and first aid, dietary and medication management, telemedicine and remote diagnosis can be delivered remotely through the Internet by connected devices [147][110][121]. Remote monitoring of patients allows more self-management of chronic conditions, and significant services improvements and cost reductions.

- *Information management*: enabled by the global connectivity of the IoT, all the healthcare information (logistics, diagnosis, therapy, recovery, medication, management, finance, and even daily activity) can be collected, managed, analysed and utilised throughout the entire value chain [56].

- *Cross-organisation integration*: the hospital information systems (abbreviated HIS) are extended to patient's home, and can be integrated into larger scale healthcare. IoT technologies facilitate the flow of patient data throughout an expanding community of care (medical centres, hospitals, nurses, physicians and associated systems) while also securing the information and protecting patients privacy [66].

The healthcare sector is just one of the markets that IoT will transform in the coming years. IoT will radical modify our medical system by bringing technology directly into the home, changing the way healthcare is delivered to patients and consumers. Moreover, with billions of heterogeneous sensors accumulating a robust network of data collection and data sharing coupled with ubiquitous identification systems, experts can conduct real-time data mining and interpretation, which leads to a continuous quality improvement to the sector.

IoT technology already offers a multitude of networked devices, cloud based applications and services for healthcare. Disparate types of healthcare data and information like logistics, diagnosis, recovery, therapy, meditation, management, finance and even daily activities (e.g., through wearable devices) can be collected from the IoT systems [56][48]. In short, more connections mean more accessible data and better healthcare for patients.

An example of the vision described above is the *Electronic Health Record* (abbreviated EHR) which is already adopted in various countries of the world [167]. Electronic Health Record is the collection and digitally storing of health information about individual's lifetime with the purpose of supporting continuity of care, education and research [94].

Healthcare IT companies are also developing *Personalised Health Records* (abbreviated PHR) where users can collect and update their facts [11], this process can be sustained and automated, by mobile based systems and IoT. IoTs automatically update physical activity, vital symptoms and similar information.

3.4 IoT Fitness Devices

In recent years, the consumer market of IoT healthcare devices has seen a proliferation of wearable fitness trackers such as Fitbit[1] and Jawbone UP[2] or smartwatches like Apple Watch[3]. These devices, along with other functions, provide a lot of health tracking features. With the rising of wearable devices people are becoming more and more interested about their health and IoT health trackers devices are becoming part of normal daily life. According to a survey conducted by the Intercontinental Marketing Services Institute for Healthcare, the sales of wearable technology will grow to almost US $30 billion by the next year[4].

IoT fitness devices can record the exercise amount, consumed energy, food intake and sleeping status of users per day. They can also measure various physical indexes such as heartbeat and respiration rate, monitor their data including speed and running distance. Sometimes they also provide support for improving exercise and goal achievements such as weight loss or distance travelled.

Being able to collect biometric data in real time for an extended period of time makes wearable devices a great tool to manage and prevent some chronic disease [130].

Fitness trackers are almost wearable devices such as smart wristbands,

[1] https://www.fitbit.com
[2] https://jawbone.com/up/trackers
[3] https://www.apple.com/watch/
[4] https://www.webcitation.org/6cxkgjwZu

heart rate strip and smart wristwatches. Some of the same functionality and sensors are also present in modern smartphones [168].

IoT fitness devices can realise exercise step counting, exercise tracking, heart rate counting, sleeping tracking as well as real-time exercise and sleeping monitoring, diet tracking, smart alarm clock, customised alarm, emotional tracking, distance course, step collection, calorie burning measurement, sleeping quality, motion reminder, smart no-sound alarm clock, distance counting and measuring calorie consumption [122][169].

Along with wearable devices, mobile phone health apps are changing the healthcare by empowering users and educating them to take control and track of their health and their fitness gains.

3.4.1 Wearable Devices

Fitness trackers have become increasingly popular during recent years. Wearable devices for fitness tracking and health monitoring consist mostly of smartwatches and wristbands. However, there are also fitness trackers that can be worn on the shoes, on the waist or on the upper arms.

3.4.2 Wristbands

A typical wrist-worn device collects and sends data such as the wearer's step count or wearer's heart rate through a gateway (e.g., a connected device like a smartphone) to the company's server.

Research has shown that data collected by these devices, despite being noisy and sometimes inaccurate, can even be used to answer intimate questions, such as whether two persons are working together (by tracking the similarity of steps per minutes between users) [174] or if the wearer has recently quit smoking [105].

All fitness trackers available in the market are equipped with an accelerometer sensor and achieve a common core functionality which is the step counting. The accelerometer sensor alone is used to infer a lot of user activities during the day such as number of steps taken, calories burned, distance travelled, as well as time slept during the night.

Wristband based accelerometers also known as *actigraphs*, such as the basic models of Fitbit or Jawbone UP, are one of the most commercially successful types of wearable sensors. Their success id due to the fact that they are cheap and can detect a wide range of daily activities (e.g., sleep, household chores, and various forms of exercises) [39][150].

Multi-sensors wristbands, as more sophisticated models of wristband, in addition to the accelerometer are also equipped with localisation services such as GPS and sensors for measuring heart rate, body temperature and blood oxygen levels (e.g., through an infrared sensor).

Data collected by wristband devices can be transmitted wirelessly for real-time feedback or uploaded to the cloud even though some basic models sync only when physically connected to a computer via cable. User interfaces of wristband devices are very limited, normally these devices are provided with only a single-button and sometimes a tiny display to show some basic information.

3.4.3 Smartwatches

A smartwatch is a wrist-worn, besides being also a timekeeping device, *"general-purpose, networked computer with an array of sensors"* [151].

Smartwatches allow more computational capabilities (actually the major part of smartwatches in the market are wearable computer) than the traditional fitness bands and host a lot of more accurate bio-sensors.

Modern smartwatches are fitted with sensors like: tri-axial accelerometers, gyroscopes, microphones, ambient light sensors, optical sensors, wireless signal strength and GPS systems.

Smartwatches' fixed mount location on the body and continual connection to the skin makes them capable of recognising wearer's physical activities with a high degree of precision. The device collocation also permits easy recording of heart rate, heart rate variability, temperature, blood oxygen, and galvanic skin response (GSR).

Reeder et al. state that smartwatches have the potential to transform the healthcare by constantly monitoring the users daily. In particular they highlight the following points of strength of smartwatches: (1) are familiar to most people, (2) are increasingly available as a consumer device, (3) enable near-real time continuous monitoring of physical activity and physiological measures, (4) support messaging and reminders, (5) enable communication between patients, family members, and healthcare providers, and (6) allow for *in situ* mini-surveys and behaviour verification based on sensors measures [152].

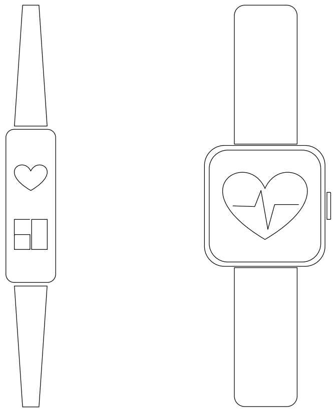

Figure 3.3: A Wristband and a Smartwatch.

Some limitations of smartwatches are the physical characteristics of the wearable devices, such as the small screen size which impaired the usability of the device [15] and the energy consumption [122] which affects significantly the batteries autonomy.

Kamdar et al. observe that improper device placement on the wrist is also a potential limitation of smartwatches for some types of sensors, for instance, heart rate can not be collected when the sensor is not in direct contact with the skin and even with skin contact variations in heart beat are observed [104]. Ahanathapillai et al. report data collection difficulties related to improper device placement when monitoring activity of elder adults[7].

Like any other wrist-worn activity device, smartwatches sometimes over-report or under-report activity levels depending on the physical characteristic of the wearer and the type of activity the wearer is doing. For instance, activities that require high levels of wrist action, such as washing hands, result in detection of increased activity level. On the contrary a wearer with limited arm movement may see an under-estimation.

3.4.4 Other IoT Fitness Devices

IoT fitness trackers ecosystem includes also some other devices that have not been mentioned in Section 3.4.3 and Section 3.4.2 which are the followings:

- *Smart scales*: electronic weight scales that measure both body weight and body fat mass and upload data wirelessly using Wi-Fi connection. Research has shown that daily self-weighing using smart scales can be effective for producing clinically meaningful weight loss [166].
- *Blood Pressure Monitors*: IoT blood pressure (abbreviated BP) monitor devices are normally composed by BP apparatus body and a communication module [96].
- *Glucose Level Monitors*: blood glucose monitoring reveals individual patterns of blood glucose changes and helps diabetic patients in the planning of meals, activities, and medication times [96]. IoT noninvasive glucose sensing solutions have been proposed in [97] and [180].
- *Oxygen Saturation Monitors*: pulse oximetry is suitable for the non-invasive uninterrupted monitoring of blood oxygen saturation. Sometimes smartwatches have this sensor integrated.
- *Body Temperature Monitors*: body temperature is one of the vital signs of a person and plays an essential role in the maintenance of homeostasis [156]. A temperature IoT measurement system is described in [103].

3.4.5 Smartphones and Health Mobile APPs

Nowadays over a billion people own a smartphone and over 40,000 medical apps have been deployed on apps marketplaces [76]. Smartphones assume a very important role in patient education, disease self-management, and remote monitoring of patients [135].

Smartphones combine mobile communication and computation into a single handheld-sised device. Modern smartphone devices available today in the market are equipped with multi-core CPUs and GPUs, megapixel cameras and high-accuracy built-in sensors such as GPS, accelerometers, gyroscopes, high resolution cameras, microphones, light s, magnetic field sensor, orientation sensors, atmospheric pressure sensors and proximity sensors.

In addition to calling and messaging features, smartphones are being used as an alternative to specialised sensors in medical devices. General purpose sensors on smartphones can detect various physiological signs of the users and can be exploited to diagnose a wide variety of medical conditions such as cough detection, irregular heartbeat detection, and lung function analysis. Smartphones can be exploited as well to perform fitness tracking task such as step counting using built-in accelerometer [28] [137][172] distance travelled, and calories burned.

Lee, Jinseok, et al. applied a technique called photoplethysmography (abbreviated PPG) to detect the heart rate from a fingertip using the built-in camera of a smartphone [117]. The same technique has been used to detect the heart rate from a recorded video stream of the patient's face [148]. Larson et al. realised a smartphone-based spirometer using the built-in microphone[115]. The user breathes near the smartphone's microphone and the sounds produced are processed by the software. Nan-Chen et al. developed a mobile smartphone-based system to detect and records nasal conditions (such as sneezing and runny nose) that occurs in everyday settings using audio from the smartphone's microphone [35]. Smartphone's camera has also been used to implement a medical app to diagnose melanoma [179].

Agu et al. highlight some benefits and challenges which derive from using smartphones as medical devices [6]; benefits of smartphones as medical devices:

- *Ubiquitous Deployment*: smartphones mobile app markets are available and accessible worldwide to billion of users which allows low costs of distribution for medical apps developers.

- *Ubiquitous Availability to Users*: users carry their mobile phone for a long period of the day. Research has shown that smartphones are in the same room as their owners for over 90 percent of the time [51].

- *Leveraging new hardware*: smartphone hardware is almost yearly updated to enhanced configurations. Medical apps, exploiting the rapid growth of computational capabilities, can run faster and better with few modifications.

Challenges of smartphones as medical devices are almost the same for smartwatches:

- *Battery Consumption*: available battery power in mobile computing applications is the most constraining resource [142]. Medical apps usually involve machine learning or image processing algorithms which are computationally intensive and quickly drain the phone's battery.

- *Noisy inputs in mobile environments*: environments (especially outdoor environments) are inherently noisy. Camera or measuring sensors can collect erroneous data in non optimal environmental conditions. For instance, outdoor lighting conditions (during a sunny day) can result in errors in heart rate monitors [145].

- *Processing complex tasks*: even though smartphones are more computationally powerful than smartwatches and wristbands, tasks that have complex memory access patterns such as machine learning are still challenging on smartphones so that offloading processing to a remote server or reducing input resolution sometimes are needed.

Chapter 4

Data Mapping

This chapter introduces the concept of Semantic Data Annotation which is the key step for every Semantic Web project. It addresses the issues which arise when mapping systems have to deal with a plethora of heterogeneous data formats and briefly reviews the main features of two of the most common data serialisations within the IoT context. Finally, the RDF Mapping Language is proposed and analysed in depth.

4.1 Semantic Data Annotation

To achieve the Semantic Web goal of making machines able to interpret, combine and use information on the Web, data need to be semantically annotated.

According to Amardeilh, *Semantic Annotation* is defined as: *"a formal representation of content, expressed using concepts, relations and instances as described in an ontology, and connected to the original resource"* [10].

Within the Semantic Web context, ontologies play a central role in annotation tasks since they explicitly define concepts and relations among them of a particular domain, in a structured and formal way.

Annotation is essentially the process of adding *metadata* to data. Metadata are *"data about data"* [85] and are normally structured according to an ontology, which means that their values refer to the instances and concepts defined in the ontology.

Consequently, semantic annotation turns human understandable content into a machine understandable form by enriching data with metadata to ensure machine readability.

It is noteworthy to underline that metadata alone without being associated to an object are meaningless.

Semantic annotation can virtually be applied to any kind of resource such as textual resources, web pages, images, multimedia contents, fields in databases and numerical data [109].

Annotations can be *embedded* or *detached*. Embedded annotations are directly added within the resource's content. Instead, detached annotations are stored outside the resource's content.

Finally, it is important that the process of semantic annotation adheres to a common standard to guarantee interoperability between different systems. The Resource Description Framework (RDF), the cornerstone of the Semantic Web, provides a standardised means for adding metadata annotations to resources.

According to Lefrançois et al. *"RDF data model may still be used as a lingua franca to reach semantic interoperability and integration and querying of data having heterogeneous formats"* [118]. Therefore, generating RDF triples (*triplify*) from sources having various formats is a key step for every Semantic Web system.

4.2 IoT Data Formats

In a context such as the Internet of Things, due to the large diversity of devices, data sources come in very large volumes and can be very heterogeneous in terms of serialisations and data formats. For instance, IoT data generated by fitness tracking devices can be normally retrieved in tabular-structured format such as CSV or hierarchical-structured format such as XML or JSON.

This section briefly describes two of the current most common data formats in the IoT fitness domain: XML and JSON.

4.2.1 eXtensible Markup Language

The *eXtensible Markup Language* (abbreviated XML) is a W3C Recommendation and a markup language for encoding data in semi-structured format [29].

XML is a metalanguage and it does not define a predefined set of tags, rather it can be used to create markup languages for specific specialised domains and purposes by specifying tags and the relationships among them.

An XML document is represented as an ordered labelled tree according to the DOM standard [184] where each node in the tree corresponds to an element and may have a value, attributes, and namespaces associated. Leaf nodes normally contain textual data values. An XML document may also carry additional element such as comments, document level information (e.g., DTD - the document type declarations), processing instructions, entities and notations.

Several XML query and processing languages are proposed and recommended by W3C such as: *XPATH* [20], *XSLT* [38] and *XQUERY* [43].

XPATH which stands for *XML Path Language* is an expression language used for navigating and selecting specific nodes within an XML document. XPATH cannot create new nodes or modifying the existing document.

Below is shown an example of a typical XML document:

```
<?xml version="1.0" encoding="UTF-8"?>
<contacts>
   <contact>
      <name>Phil Clarkson</name>
      <phone>123-456-7890</phone>
      <mobile_phone>222-654-5432</mobile_phone>
      <company>Planetgreen</company>
   </contact>
   <contact>
      <name>Adrian Vance</name>
      <phone>765-178-8236</phone>
      <company>Biolam</company>
   </contact>
   <contact>
        ...
    </contact>
        ...
</contacts>
```

An example of XPATH expression to retrieve the phone numbers of all the contacts stored could be as follows:

```
/contacts/contact/phone
```

The output returned by the XPATH expression above:

```
<phone>123-456-7890</phone>
<phone>765-178-8236</phone>
```

Note that even if the first contact has two phone numbers associated, a fixed phone number and a mobile phone number, the latter is not retrieved due to the purely syntactic approach of querying the XML tree.

4.2.2 JavaScript Object Notation

The *JavaScript Object Notation* (abbreviated JSON) is a lightweight, text-based, data interchange format [46]; is much simpler than XML and has a human-readable syntax and self-describing.

JSON was initially intended to be used in the JavaScript scripting language but then it did evolve into a language-independent data representation and it is supported by a wide range of programming languages.

JSON is essentially based on two data structures: *objects* and *arrays*. Objects are an unordered collection of name-value pairs, while arrays are an ordered list of values. JSON supports four data type which are as follows: strings, numbers, boolean expressions and null values. These features allow JSON to describe any kind of resource.

Compared to XML, JSON has higher parsing efficiency, a lighter syntax (XML is extremely verbose) and it is easier to read by humans.

Similarly to XPATH which is used to extract data from and XML document, *JSONPATH* is a declarative query language for selecting and extracting values from a JSON document [73].

The example below show the same document proposed in Section 4.2.1 serialised in JSON format:

```
{
   "contacts":{
      "contact":[{
         "name":"Phil Clarkson",
         "phone":"123-456-7890",
         "mobile_phone":"222-654-5432",
         "company":"Planetgreen"
        },
        {
         "name":"Adrian Vance",
         "phone":"765-178-8236",
         "company":"Biolam"
        },
        {
           ...
        }]
    }
}
```

4.3 Sources Heterogeneity

Sources Heterogeneity refers to when within a single domain, heterogeneous formats express homogeneous content. That is the same concepts are represented using different types and stored using a multitude of data models and formats.

A brief survey of different solutions that have been proposed for generating RDF models from data in heterogeneous formats and serialisations can be found in [53] and [54].

Dimou et al. identified some limitations of the existing mapping methods (data-to-RDF) which prevent achieving well integrated datasets: *mapping of data on a per-source basis*, *mapping data on a per-format basis* and *mapping definitions' reusability* [54].

In particular, mappings tools based on a *per-format* approach only support a specific source format (e.g, XML) which leads to a proliferation of tool to install, learn, use and maintain for each case separately or even to implement *case-specific* solutions.

Furthermore, often the mapping rules are not interoperable because they are tightly coupled to the implementation. In this case, it is not possible to reuse the mapping rules to map data that describe the same model, for different data serialisations.

Dimou et al. also proposed the requirements for generic mapping systems to address the aforementioned issues and achieve a better integration which are as follows: *uniform and interoperable mapping definitions*, *robust cross-references and interlinking* and *scalable mapping languages* [54].

In particular, the uniform and interoperable mapping definitions factor, requires the mapping definitions to be independent from the references to the input data. The same mapping definitions (i.e., mappings that capture the same concepts) should be available to be reused across different sources only by changing the reference to the specific values in the input source.

4.4 RDF Mapping Language

The *RDF Mapping Language* (abbreviated RML) is a generic mapping language which allows to map heterogeneous data sources into RDF representation [54].

From a language point of view, RML extends R2RML (RDB to RDF Mapping Language) which is a W3C recommendation for expressing customised mappings from relational databases to RDF, according to a structure and vocabulary defined by the mapping user [50].

RML, like R2RML, is a triple-oriented mapping language and can be expressed as RDF graphs and written down in Turtle syntax. However, while R2RML is specifically designed to address relational databases, RML extends this scope to a broader set of different input sources data structures and serialisations (such as CSV, XML, JSON, etc).

The main limitation of R2RML is indeed that R2RML can deal only with relational databases input.

RML, while maintaining backward compatibility with R2RML, provides a generic way for defining mappings over a wide set of heterogeneous sources adding case-specific extensions.

Given that RML, unlike R2RML, deals with different data serialisations, specific query languages are needed to refer to the content of a specific resource (e.g., XPATH for XML files or JSONPATH for JSON files).

Sources of the same domain which adapt to different structures may represent the same information and RML mapping definitions can also be re-used across them with minimal modifications and combined in a uniform way to incrementally form a well-integrated resulting dataset (see Figure 4.2.1).

Below the main structure of RML mapping graph is shortly described.

An RML mapping consists of one or more *Triple Maps*. A Triple Map is composed of three parts: (1) the *Logical Source*, (2) the *Object Map* and (3) zero or more *Predicate-Object Maps*.

The Logical Source extends the concept of a R2RML's *Logical Table* and it is used to determined the input source data to be mapped.

Reference Formulations (`rml:referenceFormulation`) are the means by which it is specified which standard or query language is used to refer to the data. The predefined Reference Formulations of the current RML version (at the time of writing) are: `ql:CSV`, `ql:XPath`, `ql:JSONPath` and `ql:CSS3`.

Unlike R2RML in which *per-row* iterations occur through the table data, the iteration pattern in RML has to be specified according to the data source format. The *Iterator* `rml:iterator` allows to define the iteration pattern over the input source and specify the extract of the data to be mapped during each iteration.

Similarly to the R2RML's property `rr:column` which defines a *column-valued term map* to determine a column's name, in RML is introduced the `rml:reference` property to reference to the single parts of the data input.

Both the iterator's value and the reference's value have to be expressed in a valid expression according to the Reference Formulation defined in the Logical Source.

The *Subject Map* (`rr:SubjectMap`) defines the criterion by which unique identifiers (URIs) are generated for the resources to be mapped. The same URIs are also used as the subject for each RDF triple produced from the Triple Map.

The *Predicate-Object Map* consists of a *Predicate Maps* and an *Object Maps*, which respectively generate the predicates and the objects for the subject generated by the Subject Map.

RML allows also cross-references through *Referencing Object Maps* which acts like a join operation between different mappings. A Referencing Object Map links together the values produced by a subject map (the parent map)

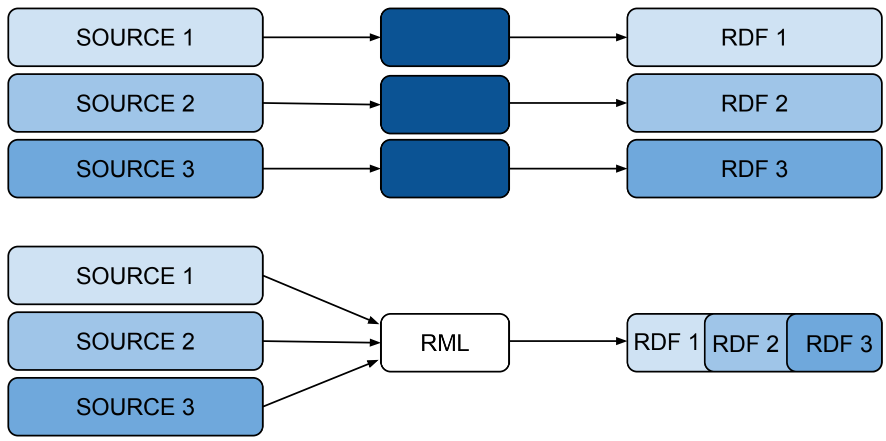

Figure 4.1: Mapping without and with RML. (Image adapted from [54]).

to the objects of triples produced by another map (the child map). The join conditions are specified by the properties `rr:parent` and `rr:child`.

RML Mapping Process

This section shortly reviews the RML mapping process and proposes a simple example of an XML-to-RDF mapping. Executing an RML mapping requires an input source and a mapping specification that describes the `TipleMaps` and points to the input source.

According to the mapping specification document, the RML processor applies the mapping rules specified in the `TipleMaps` (the `Subject Map` and the `Predicate Object Maps`) to the input data. For each point of reference to the data within the input source, values are extracted by evaluating the corresponding *target expressions* and the triples are generated.

The resulting RDF graph can be stored in a user-defined format.

Below the RML mapping definition document serialised using the Turtle syntax:

```
@prefix foaf: <http://xmlns.com/foaf/0.1/> .
@prefix rr: <http://www.w3.org/ns/r2rml#> .
@prefix rml: <http://semweb.mmlab.be/ns/rml#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#>.

<#ContactsMap>
rml:logicalSource [
    rml:source "contacts.xml";
        rml:referenceFormulation ql:XPath;
        rml:iterator "/contacts/contact";
];
rr:subjectMap [
        rr:termType rr:BlankNode;
        rr:class foaf:Person;
];
rr:predicateObjectMap [
        rr:predicate foaf:name;
        rr:objectMap [ rml:reference "name" ];
];
rr:predicateObjectMap [
        rr:predicate foaf:phone;
        rr:objectMap [ rml:reference "phone" ];
] .
```

The corresponding input source `contacts.xml`:

```
<?xml version="1.0" encoding="UTF-8"?>
<contacts>
   <contact>
      <name>Phil Clarkson</name>
      <phone>123-456-7890</phone>
      <mobile_phone>222-654-5432</mobile_phone>
      <company>Planetgreen</company>
   </contact>
   <contact>
      <name>Adrian Vance</name>
```

```
      <phone>765-178-8236</phone>
      <company>Biolam</company>
   </contact>
</contacts>
```

The RDF output graph produced:

```
_:4evc1bCWsX a foaf:Person ;
   foaf:name "Phil Clarkson" ;
   foaf:phone "123-456-7890" .

_:8yNzIbnRMw a foaf:Person ;
   foaf:name "Adrian Vance" ;
   foaf:phone "765-178-8236" .
```

In this example, data stored in the `contacts.xml` file have been semantically annotated according to the *FOAF ontology definitions* [32] and serialised in Turtle syntax.

Note that values extracted from input sources may not always be in the correct form to be directly inserted in RDF triples. RML does not provide any means for *data cleansing* and according to Dimou et al., data cleansing if necessary should be performed in advance [54]. Heyvaert et al. propose a case in which they address the problem by extending the RML vocabulary (with the terms: `rml:process`, `rml:replace` and `rml:split`) to further process the values extracted using the *regular expressions* [88].

Chapter 5

Case Study

This chapter shows how all the notions given in the previous sections have been put together to build a framework system which aims to facilitate data integration and sharing, within the context of IoT fitness devices and wellness appliances. It gives an overview of the problem and it highlights the objective and the novelty of the proposed solution. After reviewing the previous works in the literature, it illustrates the architecture of the system and the details of each one of the principal components and the design process along with the motivation behind the choices made. Finally, it presents an overview of the advanced analytic techniques for IoTs within the research field of Cognitive Computing.

5.1 Problem Overview

Nowadays, as discussed in Section 3.4, the market of the IoT fitness industry is dominated by wearable devices such as fitness trackers, smartwatches, wellness appliances and mobile health applications for smartphone.

Fitness connected devices are ubiquitous and extremely powerful since they can collect biometric data continuously and provide insights into our well-being and training sessions. For example, the Microsoft Band[1] includes an optical heart rate sensor, a skin temperature sensor, a galvanic skin response, an UV sensor, a 3-axis accelerometer/gyro, a microphone, a GPS and a barometer. Blood pressure monitors, blood glucose meters, ther-

[1]`https://www.microsoft.com/en-us/band/techspecs`

mometers and smart scales are other examples of IoT appliances in health care, which are becoming part of normal life.

IoT fitness devices also have the potential to provide health professionals a better picture of patients' overall health and fitness status. An increasing amount of health and fitness related data is constantly collected and stored in the Cloud by IoT devices.

However, from a data-centric perspective the landscape of IoT devices and wellness appliances is characterised by a high heterogeneity of representations and serialisations formats. Due to the high heterogeneity of IoT data representation formats, and the lack of common accepted standards, data remain isolated within the single systems preventing users and health professionals from having an integrated view of the various information collected. Moreover, in order to fully exploit the potential of the large amounts of data, it is also necessary to enable advanced analytics (i.e., Cognitive Computing, see Section 5.5).

All of the aforementioned considerations, do suggest the necessity of a system capable of integrating the data collected by the IoT devices, exchanging it between different platforms (especially health care information systems) without losing its meaning during this sharing and allowing analytics over collected data.

In order to achieve this goal, a lot of research in the literature suggests Semantic Web technologies as the most favourable solution approach [61] [17] [63] [84] [37] [14].

5.1.1 Related Works

Several fitness and life-log data integration solutions have already been proposed and are available.

Microsoft HealthVault[2] is a Web platform for managing and storing personal health record and fitness data. HealthVault addresses both individual and healthcare professionals and it supports several medical exchange standard formats such as Continuity of Care Record (CCR) [67]. HealthVault-enabled devices (such as Fitbit, Omron, Bayer, Sinovo) can directly upload the collected sensor measurements to the system. HealthVault provides a vocabulary to address the issues of integrating heterogeneous medical data.

[2] `https://www.healthvault.com/`

However, several lifelog data terms of the HealthVault vocabulary do not have precise definitions and their interrelationships are unclear.

Apple HealthKit[3] is a framework for integrating in a single point location data from health Apps, as well as user generated information. Apple HealthKit also provides APIs that allow third-party developers and medical sensor manufacturers to directly store their data within the Health app. Apple allows developers to define their own data types that can be stored and aggregated content can optionally be exported in XML format or encrypted and uploaded on Apple's iCloud servers for back-up purposes. On the other hand, Apps and devices which rely on HealthKit are restricted to run on iOS platforms only.

Google Fit[4] is basically the Apple HealthKit equivalent health data aggregator for Android operating systems. Google Fit is however currently limited to fitness data only while Apple HealtKit support a wider variety of health data. Google Fit aggregated content is accessible via the Web portal or through a REST (REpresentational State Transfer) API. Google Fit defines fixed set of data types which can be stored, third-part developers need to inform Google to add and share new ones.

Google Fit and Apple HealthKit are intended to be data aggregators for their respective ecosystems and lets health and fitness applications as well as wearable devices gather health information in one single point location. Google Fit is currently limited to fitness data only while Apple HealtKit support a wider variety of health data. However, data remain confined to their respective platforms.

MyFitnessCompanion [72] is a health and fitness app which enables users to aggregate their data in one place in a similar way to Apple HealthKit and Google Fit. MyFitnessCompanion aims to integrate off-the-shelf, commercially available devices, it can interacts with a wide range of wireless devices and wearable health trackers, and also aggregates data from third-party apps. It connects with Microsoft HealthVault, Google Fit, Fitbit, Withings, Jawbone, and iHealth servers and other EHR systems. However, MyFitnessCompanion can be used only on an Android platform.

MELLO [107] is an ontology for representing health-related life-logging data including definitions, synonyms, and semantic relationships. MELLO fills the semantic gap between heterogeneous lifelog terms that are generated

[3]`https://developer.apple.com/healthkit/`

[4]`https://www.google.com/fit/`

by diverse health self-tracking devices. The unified representation of lifelog terms facilitated by MELLO can help describe an individual's lifestyle and environmental factors, which can be included with user-generated data for clinical research and thereby enhance data integration and sharing. However, MELLO does not make use of Semantic Web technologies.

HealthIoT [153] is an OWL ontology which aims to neatly and comprehensively represent the harmonisation between the medical IoT domain knowledge and the healthcare domain knowledge. HealthIoT integrates upper level ontologies such as SSN (Semantic Sensor Network Ontology) [40] to model concepts about acquisition sensors and Time Ontology [45] to model time concepts. As described by Rhayem et al. in [153], HealthIoT has been used within IoT Medicare system, which is an intelligent decision support system integrated with query and inference engine based on Semantic Web technologies. However, HealthIoT only partially covers the domain considered in this thesis, it doesn't model higher abstract fitness and wellness related concepts such as running or other sport training sessions or meditation sessions. The authors also do not clearly specify which kind of data formats and serialisations are currently supported among commercial health IoT devices. Moreover, HealthIoT ontology is not currently publicly available.

Even though some of the projects listed here rely on vocabularies or ontologies to achieve their purpose, all of them do not make use of Semantic Web technologies for their implementation, with the only exception of HealthIoT.

5.1.2 Objective and Novelty

The objective of this thesis is to design and implement an ontology based system to (1) allow data interoperability among heterogeneous IoT fitness and wellness devices, (2) facilitate the integration and the sharing of information and (3) enable advanced analytics over the collected data. The novelty of the solution proposed lies in exploiting Semantic Web technologies to formally describe the meaning of the data collected by the IoT devices and define a common communication strategy for information representation and exchange.

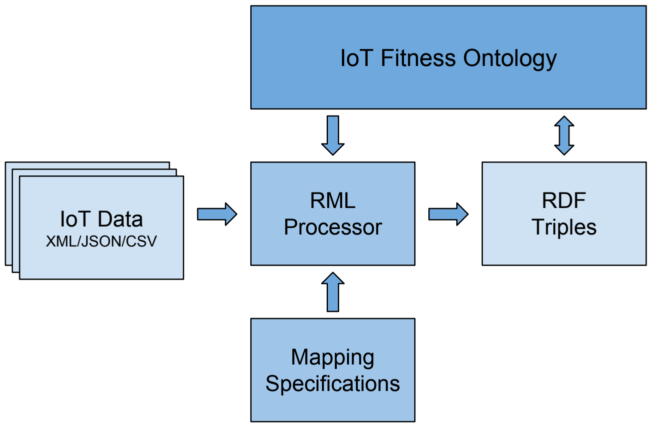

Figure 5.1: Overall architecture of the proposed system.

5.2 System Architecture

Semantic data annotation is the key step for every Semantic Web project (Section 4.1). The framework proposed in this thesis project aims to semantically annotate heterogeneous IoT fitness and life logging data collected by wearable devices and wellness appliances in order to make it integrated and machine-understandable. This way, healthcare, wellness and fitness programs can process the data according to the defined meaning with the aid of semantic rules and inference engines to provide actionable knowledge and more effective solutions. In this project Semantic Web technologies are used to define the data collected from the IoT devices and sensors, ensuring consistency in the related terminology. Moreover, Semantic Web technologies can also be used to define rules about IoT-based healthcare services and fitness programs.

Figure 5.1 shows the overall architecture of the proposed framework. The two core components of the entire system are the ***IoT Fitness Ontology*** (abbreviated IFO) and the *mapping process* (i.e., the RML processor and the mapping specifications).

The primary role of the IFO ontology is to provide a formal representation of the main concepts within the IoT fitness domain.

The ontology is an essential component to achieve interoperability, analyse, integrate, store and transfer IoT data in the most accurate and secure way. The IFO ontology, developed within this project, describes the data not only by its measurement value, but also its relationships with other data sources and also with descriptive properties like where and when it was produced. More importantly, the IFO ontology also relates the described concepts to health data standards domain ontologies like SNOMED-CT.

The RML processor, supplied with the mapping specifications for the various sources, consumes the IoT raw data and transforms it into an RDF graph, which is the same input data semantically annotated according to the IFO ontology.

After the IoT fitness devices data is semantically annotated, the health care services can provide proper insights about by using the inference and role engines (or other advanced techniques) that are offered by semantic Web technologies to achieve useful actionable knowledge, thus exploiting the intrinsic data potential to its maximum.

The next sections illustrate in details the design process and the characteristics of the IFO ontology and the mapping process along with the motivation behind the choices made.

5.3 IoT Fitness Ontology

The IFO ontology is one of the two core components of the system designed in this thesis project. This section outlines the development process adopted in order to design the IFO ontology and secondly it gives an accurate description of the ontology structure including details.

5.3.1 Design Process

The IFO ontology, as already mentioned in Section 5.1, is the main core component of the system object of this thesis project. The ontology aims to represent the most common and important of concepts within the domain of the IoT fitness devices and wellness appliances.

The characteristics and functionality provided by several IoT wearable devices and wellness appliances, as well as health mobile applications available in the market, were considered and carefully analysed in order to identify the concepts described in the IFO ontology. The list of products

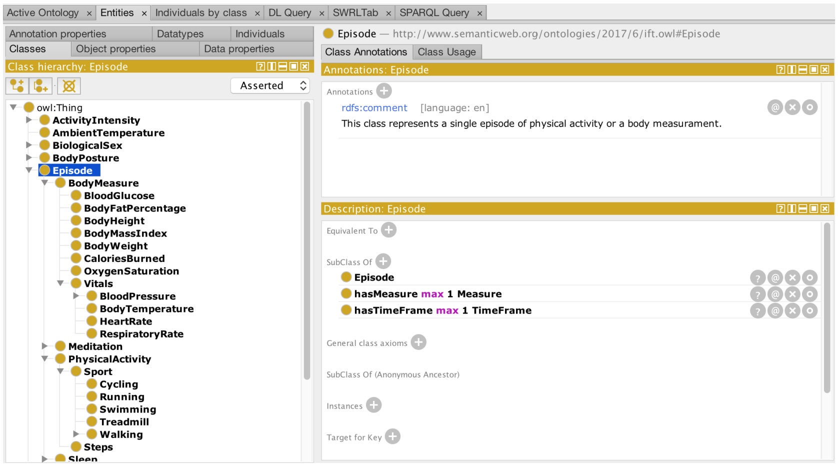

Figure 5.2: Protegé screenshot.

and vendors that were taken in consideration during the design process includes: Apple HealthKit, Microsoft HealthVault, Google Fit, Fitbit, Jawbone, Strava, Runtastic, iHealth and Nokia Health.

Tool and Techniques

The ontology was written using OWL language and modelled with Protegé as ontology-editing environment. Protegé was chosen since it is the most widely used open source ontology editor available to the OWL community and it enables the creation and representation of ontologies in OWL using a visual editor and in addition it support automated reasoning tasks such as consistency checking and automatic classification of classes using description logic expressions [140]. The ontology has also been validated using using the ontology reasoner HermiT [158] to check for inconsistencies integrated as default reasoner system in Protegé 5.2.

Figure 5.2 shows the IFO ontology classes hierarchy as can be seen within the Protegé editor.

Knowledge-Engineering Methodology

The methodology proposed by Noy and McGuiness [139], which is a simple but complete knowledge-engineering methodology to build ontologies, was adopted as development model. The process consists of seven steps which are as follows:

1. *Determine the domain and scope of the ontology*: the domain of interest is the set of IoT fitness wearable devices and wellness appliances, including mobile fitness and health applications as shown in Section 3.4.

   The scope of the ontology is to provide a formal and machine-readable representation of the main and most common concepts within the domain of interest and the relationships to each other.

2. *Consider reusing existing ontologies*: to achieve a better integration with other systems and better specify the meaning of each class, references to other standardised ontologies such as SNOMED-CT were made. Personal information (e.g., date of birth) was based on FOAF ontology and the Basic geo (WGS84 lat/long) vocabulary was used for the geospatial locations. In order to keep the ontology simple, the import of other OWL top level ontologies for concepts related to the measurements (e.g., units of measure) or for the time information (e.g., time intervals) has been avoided.

3. *Enumerate important terms in the ontology*: the key terms used in the ontology are the nouns describing generic types of physical activities and physiological parameters with no relation to specific brands with the exception for the devices classification part. Examples of terms used about physical activities are: *Steps, Running, Walking, Swimming, ActivityIntensity, FlightsClimbed.*

   Examples of terms used about physiological parameters are: *HeartRate, BodyTemperature, BodyWeight, BloodPressure, CaloriesBruned.*

   Other general terms are: *Meditation, TemporalRelationship, BodyPosture, Measure, Statistics, TimeFrame, MassUnit.*

4. *Define classes and the class hierarchy*: according to Uschold and Gruninger there are several possible approaches in developing a class hierarchy [175] such as: *top-down*, *bottom-up* or a combination of these two. For the IFO ontology was mainly used the top-down approach. The ontology is build around the root class *Episode* which represent

the set of the all possible events that can be measured by the IoT devices and wellness systems (a detail explanation of the concept of episode is given in Section 5.3.2).

5. *Define the properties of classes and slots*: properties have been defined to model the relationships among concepts. The most important object properties relate an episode to its measure (OWL property `hasMeasure`) and to its time reference (`hasTimeFrame`).

   Some concepts regard only specific concepts such as `BodyPosture` makes sense only if connected to the concept of `BodyMeasure`.

6. *Define the facets of the slots*: in this step cardinality constraints and value restrictions were defined.

   For instance, a single episode cannot have multiple measurements or multiple time references directly associated so a maximum cardinality restriction of 1 has been asserted for this class.

   It is noteworthy to underline that some measurements require more than a single numerical value such as the blood pressure. The blood pressure is measured in millimetres of mercury (mmHg) and is written as two numbers (e.g., 120/80mmHg). The first (120 in the example aforementioned) number is the systolic blood pressure and the second number (80) is the diastolic blood pressure. Systolic blood pressure and diastolic blood pressure according to the IFO ontology are two separated events.

7. *Create instances*: units of measurement or commercial devices were modelled as OWL individuals since are concepts that cannot be specialised anymore in the hierarchy.

The result is a harmonised ontology of the most important common concepts in the domain considered. The first version of the IFO ontology consists of 93 classes, 16 object properties, 7 data properties, and 47 individuals.

5.3.2 Ontology Structure

The IFO ontology is built around the notion of *Episode*. An episode represents the set of the all possible events that can be measured by the IoT devices and wellness systems. For example, an episode could be the heart rate measured during a running training session by a wearable wrist worn heart rate monitor or the person's body weight measured by a smart scale. To each episode is associated a time reference and a numeric measurement value with the related unit of measurement. The time reference can be a single point in time or a time interval, that is, the start time and the end time of the event. These information are essential because they allow to numerical quantify the object of the event and give it a temporal collocation and duration (Figure 5.3).

Within the IFO ontology, the concept of episode is modelled by the OWL class `Episode`, which is also the root class of the entire episodes hierarchy, all the other concepts inherit the properties associated with it. The IFO ontology organises episodes in a hierarchical structure based on single inheritance. Along the hierarchy two main categories of episodes can be distinguished: (1) the physical activities and (2) the body measurements. Physical activities comprehend any kind of activity involving body movement such as walking, running, swimming or steps taken. Body measurements, on the other hand, are relative to the physiological parameters of a person such as the body weight or body height or the person's vital signs such as the heart rate or the blood pressure. Other minor categories of episodes that the IFO ontology defines, concern the sleep and the meditation.

Other fundamentals components of the IFO ontology are the OWL class `Measure` and the class `TimeFrame` which they respectively model the measurement and the time reference; these two classes are associate to the `Episode` class through the OWL properties `hasMeasurement` and `hasTimeFrame` as shown in Figure 5.4.

Furthermore, the IFO ontology also includes supplementary classes which describe concepts that can be used in addition to the fundamental ones which are as follows: `UserNote` which is about user personal annotations about an episode, `InputSource` that is the kind of device (e.g., wearable device or a smartphone) by which the measurement has been made, `geo:Point` for geolocation information relative to the episode (this can also be use to represent information about a path taken during an outdoor training session), `TemporalRelationship` temporal relationships with respect to other person's life activities (e.g., temporal relationship of an episode with respect to meals), `Statistics` to give more information about the numerical value of the measurement (e.g., the measurement is an average or the maximum

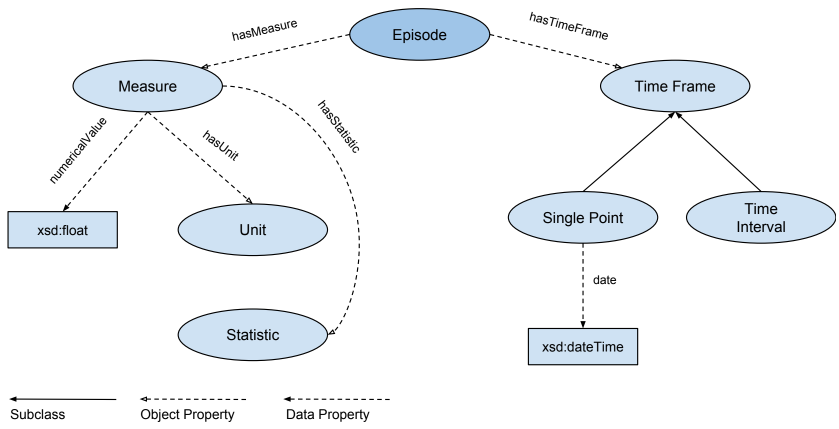

Figure 5.3: Excerpt of IFO ontology graph: the Episode concept.

or the minimum value among several values), `ActivityIntensity` for the physical activity intensity (i.e., light, moderated or vigorous), `BodyPosture` the person's body posture taken during the measurement and personal information such as the gender or the date of birth of the user.

Units of several measurement systems are modelled by OWL individuals and are instances of their respective OWL classes organised depending on their functions. For example, the OWL individuals `kg` and `lb` which respectively represent kilogram and pounds are instances of the class `MassUnit` which is, in its turn, a subclass of `Unit`.

Devices used to acquire data about an episode are represented in the IFO ontology by the class `InputSource` and are classified in `Wearable` for wearable devices, `Appliance` generic systems, `Smartphone` for mobile applications and `UserTyped` for episodes recorded manually by the user.

5.4 Mapping System

Along with the IFO ontology, the mapping process constitutes the second core component of the proposed framework.

From a data perspective, the context of IoT, is characterised by a high heterogeneity of data representation and serialisation formats. Among dif-

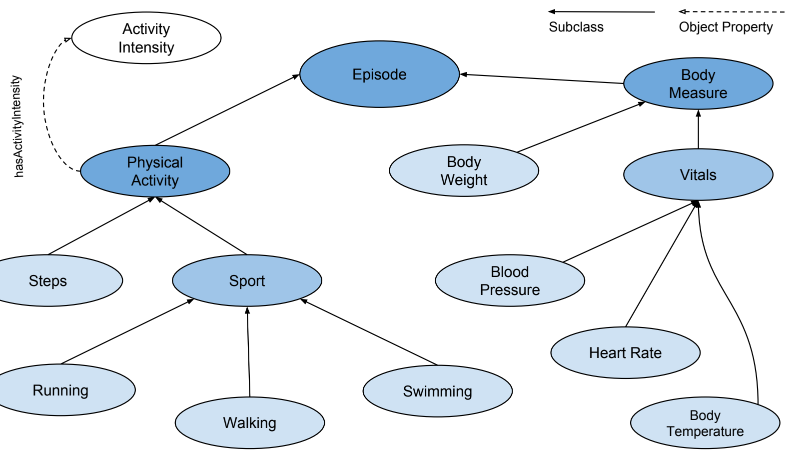

Figure 5.4: Excerpt of IFO ontology: the Episode hierarchy.

ferent vendors of IoT fitness devices the same concepts are represented using different types and stored in different formats.

The mapping system was implemented using RDF Mapping language (RML) along with the RML Processor since RML allows mapping definitions that can be reused across different implementations for different source formats reducing implementation costs.

Mapping specifications were defined for three IoT systems among the ones that have been used to construct the ontology (i.e., Fitbit, Apple Health and Nokia Health). In particular, the mapping rules are relative to some shared concepts among these systems (e.g., the heart rate). As an evidence of the flexibility of the mapping language, the aforementioned IoTs devices were selected because they use different formats to store the data collected. Even though only a limited number of devices were selected, mapping definitions can be easily reused across different sources that provide similar information. As a mapping process executor, RML Mapper was used. RML Mapper which is a Java implementation of an RML mapping processor. RML Mapper already supports XML, JSON and CSV data formats, and therefore there was no need to extend or modify the existing software.

The next sections present a brief overview of how data can be retrieved from the IoT fitness devices and the most common data serialisations used by them. Three different data excerpts are shown as example of the high heterogeneity that characterises the IoT domain. Finally, it discusses in details the mapping specifications.

5.4.1 IoT Data

Raw data collected by IoT devices can be manually retrieved when systems are provided with data export functionality (e.g., Apple Health). IoT raw data can normally be exported in XML or CSV serialisation formats. On the other hand, when a data export function is not directly available within the device or the mobile application, data collected by IoT systems can be downloaded from the Cloud, usually in JSON format, through RESTful APIs provided by the device vendor (e.g., Fitbit).

Data Excerpts

Fitbit users can retrieve data collected by activity trackers and smart scales, from the Cloud, using web APIs provided by the vendor.
The JSON code below is the response obtained after executing an HTTP GET request, after being authenticated and authorised to the Fitbit server[5]:


```
{
    "weight":[
        {
            "bmi":23.57,
            "date":"2015-03-05",
            "logId":1330991999000,
            "time":"23:59:59",
            "weight":73,
            "source": "API"
        },
        {
            "bmi":22.57,
            "date":"2015-03-05",
```


[5] https://dev.fitbit.com/reference/web-api/body/

```
            "logId":1330991999000,
            "time":"21:10:59",
            "weight":72.5,
            "source": "Aria"
        }
    ]
}
```

In the above example, the output given consists of a list of all user's body weight log entries for a given day using units in the unit measurement system which corresponds to the *Accept-Language* HTTP header provided during the request. The specific device by which the data have been collected, date and time, and the numerical value of the measurement are all specified within the response.

The next example shows body weight data collected by Nokia Health smart scale. The output is in CSV format and has been obtained using the export function provided on the Nokia Health online dashboard[6]:

```
Date,Weight,"Fat mass","Bone mass","Muscle mass",Comments
"2017-08-10 20:31:00",82.00,10.00,,,,
"2017-08-07 11:10:50",81.00,,,,,
```

In addition to the body weight the CSV file might also contain information about the fat mass, the bone mass, the muscle mass and user personal comments. It is noteworthy to highlight that the date and time are stored within a unique string while in the Fitbit example they were separated.

Below is shown an excerpt of data manually exported from Apple Health[7] in XML format:

[6] https://health.nokia.com
[7] https://www.apple.com/lae/ios/health/

```
<Record type="HKQuantityTypeIdentifierBodyMass"
        sourceName="Lifesum"
        sourceVersion="6.2.0.7"
        unit="lb"
        creationDate="2016-06-08 16:47:26 -0400"
        startDate="2016-06-08 00:00:00 -0400"
        endDate="2016-06-08 00:00:00 -0400"
        value="150"
/>
```

Once again, information is about the body weight of the user but in this serialisation also the unit of the measurement and the data source (that is a mobile application) are included.

The excerpts above are straightforward examples of the issues related to the heterogeneity of data representation and serialisation formats used within the IoT fitness domain as discussed in Section 4.3. The same concept of *body weight* is represented in three different ways and serialised in three different formats.

5.4.2 Mapping Specifications

In order to semantically annotate IoT data according to the IFO ontology, that is translating input data into an RDF graph, the RML processor requires mapping specifications for the various data targets. However, the RML language allows mapping definitions that can be reused across different implementations for different source formats reducing implementation costs.

As illustrated in Section 4.4, RML mapping specifications are based on one or more *Triples Maps* which define how the triples (the resulting RDF graph) are generated. Essentially a triple map contains a rule to generate zero or more RDF triples which share the same subject for each extract of data from the input source. A single triples map is composed by the *Logical Source*, the *Subject Map* and zero or more *Predicate-Object Maps*.

As an example, below, is analysed the triples map used to generate the RDF graph starting from the Fitbit data about the body weight. For convenience, here, is proposed again an excerpt of the data input source (the same example shown in Section 5.4.1):

```
{
    "weight":[
        {
            "bmi":23.57,
            "date":"2015-03-05",
            "logId":1330991999000,
            "time":"23:59:59",
            "weight":73,
            "source": "API"
        }
    ]
}
```


Below the triples map `<#FitbitBodyMass>` used to map the example data above into RDF triples.

```
<#FitbitBodyMass>
rml:logicalSource [
        rml:source "fitbitWeight.json";
        rml:referenceFormulation ql:JSONPath;
        rml:iterator "$.weight";
];

...
```

The logical source consists of the reference to the input source to be mapped, in this case the `fitbitWeight.json` file. The *Reference Formulation*, pinpoint by `rml:referenceFormulation`, specifies how references to the data occurs and, since RML uses references relevant to the input source, in this case JSONPath is used. The iterator specifies how to iterate over the input data, here is specified by the JSONPath expression: `$.weight`.

```
...

rr:subjectMap [
        rr:termType rr:BlankNode;
        rr:class fo:Measure;
];

...
```

The subject map consists of the template that defines the URI pattern used to generate the subject of the triple and optionally its type. In this case a blank node is generated and the triple is typed as `fo:Measure`; `fo` is the name space used for the IFO ontology.

```
...
rr:predicateObjectMap [
        rr:predicate fo:hasNumericalValue;
        rr:objectMap [
                rml:reference "@.weight";
                rr:datatype xsd:float;
        ];
];
...
```

A *Predicate Object Map* consists of a *Predicate Map* that specifies the predicate of the triple and an *Object Map* which specifies the object (one or more) of the triple. Specifically in this case a JSONPath expression is used to point to the body weight value in the source (`rml:reference "@.weight"`).

The resulting RDF graph:

```
_:kWRuix2ft9 a fo:BodyWeight ;
        fo:hasMeasure _:fxbMJQzZG8 ;
        fo:hasTimeInterval _:CrHFdBYBD8 .

_:fxbMJQzZG8 a fo:Measure;
        fo:hasNumericalValue "73"^^xsd:float ;
        fo:hasUnit fo:kg .

_:CrHFdBYBD8 a fo:TimeInterval ;
        fo:endDate "2015-03-05 23:59:59"^^xsd:dateTime ;
        fo:startDate "2015-03-05 23:59:59"^^xsd:dateTime .
```

Below, an excerpt of the mapping specification for the Apple Health body weight data example (as shown in Section 5.4.1):

```
<#HKBodyMassMeasure>
rml:logicalSource [
   rml:source "export.xml";
   rml:referenceFormulation ql:XPath;
   rml:iterator "/HealthData/
         Record[@type=\"HKQuantityTypeIdentifierBodyMass\"]";
];
rr:subjectMap [
   rr:termType rr:BlankNode;
   rr:class fo:Measure;
];
rr:predicateObjectMap [
   rr:predicate fo:hasNumericalValue;
   rr:objectMap [
   rml:reference "@value";
   rr:datatype xsd:float;
   ];
];
```

```
rr:predicateObjectMap [
   rr:predicate fo:hasUnit;
   rr:objectMap [
   rr:template "http://www.fitnessontology.com/#{@unit}";
   ];
].
```

Compared to the Fitbit mapping specification, since it is an XML file, XPATH has been used instead of JSONPath and as can be seen in the above example only the target expressions have been changed.

5.5 IoT Data Analytics

The next real challenge in the IoT landscape will be to make the collected data meaningful and useful.

According to Sheth the next step in evolution of IoT data will involve more advanced data analytics, that is, integrated and knowledge-enhanced analytics of IoT data which implies a shift from raw data processing to more intelligent data processing [159].

Advanced applications of IoT systems, involve a broad variety of different sensors, which imply different modalities for collecting data of interest. Moreover, IoT data is often complemented by social and Web data, collective intelligence and curated knowledge (i.e., ontologies). Inevitably, IoT applications will face the classic big data problems in more extreme forms: increasing volume, broader variety, increasing complexity, rapid changes, and more veracity challenges, encompassing trust, security, and privacy.

Within the research community, a *smart IoT*, is an IoT ecosystem that supports making sense of all the IoT big data. Smart IoT ecosystems enable intelligent applications which provide higher-quality and timely decisions making and actions.

As Sheth explains, IoT intelligent data processing implies converting massive amounts of raw data into something which is contextually relevant or meaningful for situational awareness, decision making and taking actions [159].

In Figure 5.5, Sheth shows an example of how starting with a sensor reading, actionable data can be derived according to the *Data-Information-*

Knowledge-Wisdom Hierarchy (abbreviated DIKW) originally proposed by Ackoff [3] which is a widely recognised and *taken-for-granted* model in the information and knowledge literature [155].

The lowest level concerns sensor and device data and it shows "150" which is a blood pressure measurement. The second level concerns semantically annotated data or information. The third level concerns knowledge; in the example, based on the health guidances used by clinicians, it shows a medical condition of *"elevated blood pressure"*. The top layer concerns wisdom, in fact the elevated blood pressure alone, is not an actionable information: the clinician has to find out whether this is due to hyperthyroidism or hypertension in order to prescribe a proper medication [159].

The DIKW pyramid shows that data can be used to create information; information can be used to create knowledge, and knowledge can be used to create wisdom.

According to Sheth, the process to derive wisdom from physical, cyber and social big data involves the use and synergy of three computing paradigms, that is, *Semantic Computing*, *Cognitive Computing* and *Perceptual Computing* [159].

Semantic Computing

Semantics is essentially about associating meaning with data. *Semantic Computing* allows to deal with data in context despite differences in syntax and representation structure.

Semantics has been widely discussed in Section 4.1.

Cognitive Computing

Cognitive Computing is an emerging field characterised by a synergistic confluence of cognitive science, data science, and a multitude of computing technologies [93].

According to Kelly Cognitive Computing systems are: *"systems that learn at scale, reason with purpose and interact with humans naturally. Rather than being explicitly programmed, they learn and reason from their interactions with us and from their experiences with their environment* [106].

A cognitive system interprets data by learning in a way that loosely mimics the process of human mind cognition. Cognitive systems make use of data mining techniques, machine learning algorithms, neural networks,

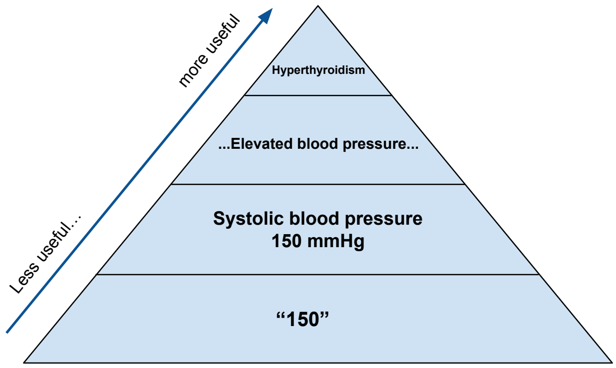

Figure 5.5: From data to decisions and actions: climbing the Data, Information, Knowledge, and Wisdom (DIKW) ladder. (Image adapted from [159]).

deep learning, reasoning, natural language processing, information retrieval, big data, cloud computing, IoT systems, speak recognition and computer vision along with other various artificial intelligence techniques for analysing a massive amount of data in order to support humans to make decisions and answer complex questions.

Cognitive Computing systems are different from traditional computing systems. Cognitive Computing systems do not use brute force approaches, instead they are adaptive, they learn and evolve over time, and incorporate context into the computation. Cognitive Computing systems sense their environment, think and act autonomously, and deal with uncertain, ambiguous, and incomplete information.

A notable example of Cognitive Computing system is IBM Watson [68]. Watson is also the first Cognitive Computing system that leverage the synergy between cognitive science and an array of computing technologies. In 2011 Watson performed at a level to win *Jeopardy!*[8] game against the two all-time human champions.

[8] `https://en.wikipedia.org/wiki/Jeopardy!`

Perceptual Computing

According to Sheth et al. *Perceptual Computing* in its simple form: *"is founded on rich domain knowledge that connects causes with effects and on reasoning strategies that can predict the effects of causes and explain the effects using causes"* [160].

Semantic Computing refers to interpreting sensed data, in order to build a model of the current situation (context), dealing with incomplete or ambiguous information.

Within IoT context, deriving an abstraction (i.e., a representation of an environment) based on an incomplete set of observations from the physical world is formulated as an iterative abductive-deductive reasoning process [87].

5.6 Cognitive Internet of Things

The concept of *Cognitive Internet of Things* (abbreviated CIoT) extends the idea of Cognitive Computing to IoT.

According to Wu et al. Cognitive Internet of Things can be defined as: *"a new network paradigm, where (physical/virtual) things or objects are interconnected and behave as agents, with minimum human intervention, the things interact with each other following a context-aware perception-action cycle, use the methodology of understanding-by-building to learn from both the physical environment and social networks, store the learned semantic and/or knowledge in kinds of databases, and adapt themselves to changes or uncertainties via resource-efficient decision-making mechanisms"* [185].

Existing IoT applications are still highly dependent on human beings for cognition processing whereas in Cognitive Internet of Things, smart objects behave as agents, and interact with physical environment and social networks with minimum human intervention.

Cognitive Internet of Things enhances the current Internet of Things by mainly integrating the human cognition process into the system design.

The opportunity to leverage the IoT with cognitive computing together provides great possibilities in healthcare [176].

Sheth et al. developed a framework for continuous monitoring of patients by collecting large quantity of physical-cyber-social and medical data with the intention of converting data into actionable information to make timely medical decision exploiting Semantic Computing, Cognitive Computing and Perceptual Computing [160].

IBM Watson Health is one of the most well-known examples of the integration of big data and machine learning to help leverage value from IoT data within healthcare domain [1], including sport analytics[9].

Wu et al. propose a an operational framework for CIoT, which mainly characterises the interactions among five fundamental cognitive tasks: perception-action cycle, massive data analytics, semantic derivation and knowledge discovery, intelligent decision-making, and on-demand service provisioning [185].

Gyrard et al. [83] suggest an architecture for CIoT systems which is largely divided into three layers by their functions: (1) the *physical layer* which add semantics to data to unify them, by using semantic web languages (such as RDF, RDFS, OWL) and domain ontologies, (2) the *virtualisation layer* which mainly infers high level knowledge using reasoning engines performed on data and by exploiting the web of knowledge available online and (3) the *cyber layer* which allows developers to build large scale and meaningful IoT applications on top of the virtualisation layer and reduces the IoT application development, thus enabling rapid prototyping and encourage interoperability of services.

[9] `https://www.ibm.com/internet-of-things/iot-zones/sports-analytics`

Chapter 6

Conclusions

This chapter summarises the main contributions of this thesis project and outlooks several possible directions for improvements and related future works. Finally, it discusses some considerations regarding the social impact of this work.

6.1 Discussion

The IoT fitness devices and wellness appliances domain is characterised by a high heterogeneity of data representation and serialisation formats and lacks of common accepted standards. These interoperability issues cause a confinement of the collected data which remains isolated within each single system, preventing users and healthcare professionals to have an integrated view of the information and data acquired. Several solutions have been already proposed both from industry and academia. However, due to trade policies, commercial data integrating systems do not allow data exchanging among different systems. In the other cases, reasoning prospects have not been taken in consideration as the main feature and none of the works revised make use of Semantic Web technologies.

In this thesis project, a Semantic Web approach has been adopted to design and develop an ontology based system to allow data interoperability among heterogeneous IoT fitness and wellness devices, facilitate the integration and the sharing of information and enable advanced analytics over the collected data. The proposed system allows the healthcare services and fitness programs to provide proper insights by using the inference and role engines (and other advanced techniques) that are offered by Semantic Web technologies in order to achieve useful actionable knowledge, thus exploiting the intrinsic IoT health and fitness data potential to its maximum.

6.2 Future Works

This section recommends some several minor improvements which could be made to the existing system and discusses the future challenges.

6.2.1 Minor Improvements

Minor improvements comprehend slight adjustments and further extensions to the system to enhance the actual functionality.

Adding More Mapping Specifications

In order to test the framework, mapping specifications have been written only for three IoT systems, a wider mapping coverage is needed.

Extending the Current Ontology

Even though the current version of IFO ontology covers over 85% of the IoT fitness devices and wellness appliances concepts, specific vendor devices concepts have been deliberately excluded. Nutrition, anaerobic activities and drug management concepts could constitute a considerable and noteworthy extension to the ontology.

Data Cleaning

Data cleaning is the process of detecting and correcting corrupted, inaccurate or incorrect values withing IoT raw data. Data cleaning is outside the scope of this project. The implementation of a data cleansing stage is highly recommended.

System Evaluation and Comparing Tests

Quantitative and qualitative tests for the ontology, especially quality evaluation and comparing tests of the framework to similar systems have not been made but are suggested.

6.2.2 Future Challenges

This section proposes more significant extensions to the actual system and suggests the direction for a possible practical application.

Implementation of a Reasoning System

Integrating and sharing data is not enough, the added value is to interpret the data in order to achieve actionable knowledge. Reasoning systems are an essential part to fully exploit the potential of IoT health and fitness data and Cognitive Computing (e.g., inference engines) seems a promising way to achieve the goal.

However, due to the complexity of the healthcare domain, reasoning models and inference rules cannot be re-invented or re-designed each time, a mechanism for *"sharing reasoning"* is needed.

An example of this concept can be founnd in [82] where Gyrard et al., suggest the concept of *Linked Open Rules*. Stemming from Linked Open Data movement, Linked Open Rules allow exploiting, reusing and combining rules to help developers design and combine cross-domain IoT applications.

Integration of the System with OWL Upper Level Ontologies

In order to better support advanced reasoning over data, especially rule based reasoning, an integration with other standard OWL upper level ontologies is essential.

Suggested ontologies for concepts related to time, units of measurement and sensors are as follows:

- *QUDT*: was originally developed for the NASA. The QUDT ontology is designed to provide comprehensive coverage of almost every unit of measurement [90]. This kind of information are particular important because numeric data without any formalised units is pretty useless for machines and IoT fitness heavily rely on this kind of concepts.

- *OWL-Time*: is ontology of temporal concepts, and temporal properties and a W3C recommendation. The ontology provides a vocabulary for expressing facts about topological (ordering) relations among instants and intervals, together with information about duration, and about temporal position including date-time information [45].

- *Semantic Sensor Network Ontology*: is a W3C recommendation and an ontology for describing sensors and their observations, the involved procedures, the studied features of interest, the samples used to do so, and the observed properties [40].

An integration with an OWL version of SNOMED-CT is also highly recommended.

Integration of the IFO Ontology with Existing Healthcare Informative Systems

Currently, there has not been implemented an actual integration of the framework with a private or a government controlled EHR system. Future works in this direction should rely on standards for medical information exchange such as HL7 [119].

Health Level Seven International (abbreviated HL7) is a non-profit organisation for providing a comprehensive framework and related standards for the exchange, integration, sharing, and retrieval of electronic health information that supports clinical practice and the management, delivery and evaluation of health services[1].

Design a Linked Open Data Web Portal for Sharing and Storing Health and Fitness data

A practical application of the proposed system could be the development of a Linked Open Data web portal to provide IoT fitness data sharing and support the scientific research [171] on the model of the Open Human Project[2].

The Open Human Project aims to let individuals access and share their data with researchers. To this end, the Open Humans Network created an online system that helps match people who want to share their health and fitness data with researchers who would benefit from access to information.

[1] http://www.hl7.org
[2] https://www.openhumans.org

For the design of the web portal, as triple store and underlying framework is recommended the Jena framework. Jena is a free and open source Java framework for building Semantic Web and Linked Data applications [102].

Bibliography

[1] Ibm opens watson iot global headquarters, extends power of cognitive computing to a connected world. *press release*, 2015. http://www-03.ibm.com/press/us/en/pressrelease/48443.wss.

[2] The open definition., 2015. `http://opendefinition.org`.

[3] R. L. Ackoff. From data to wisdom. *Journal of applied systems analysis*, 16(1):3–9, 1989.

[4] B. Adida, M. Birbeck, S. McCarron, and S. Pemberton. Rdfa in xhtml: Syntax and processing. *Recommendation, W3C*, 7, 2008.

[5] I. D. Agranat. Engineering web technologies for embedded applications. *IEEE Internet Computing*, 2(3):40–45, 1998.

[6] E. Agu, P. Pedersen, D. Strong, B. Tulu, Q. He, L. Wang, and Y. Li. The smartphone as a medical device: Assessing enablers, benefits and challenges. In *Sensor, Mesh and Ad Hoc Communications and Networks (SECON), 2013 10th Annual IEEE Communications Society Conference on*, pages 76–80. IEEE, 2013.

[7] V. Ahanathapillai, J. D. Amor, Z. Goodwin, and C. J. James. Preliminary study on activity monitoring using an android smart-watch. *Healthcare technology letters*, 2(1):34–39, 2015.

[8] H. Alemdar and C. Ersoy. Wireless sensor networks for healthcare: A survey. *Computer Networks*, 54(15):2688–2710, 2010.

[9] H. T. Alvestrand. Tags for the identification of languages. 2001.

[10] F. Amardeilh. Semantic annotation and ontology population. *Semantic Web Engineering in the Knowledge Society*, page 424, 2008.

[11] A. A. Atienza and K. Patrick. Mobile health. *American journal of preventive medicine*, 40(5):S151–S153, 2011.

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

[13] F. Baader. *The description logic handbook: Theory, implementation and applications*. Cambridge university press, 2003.

[14] C. J. Baker and K.-H. Cheung. *Semantic web: Revolutionizing knowledge discovery in the life sciences*. Springer Science & Business Media, 2007.

[15] M. Bang, K. Solnevik, and H. Eriksson. The nurse watch: design and evaluation of a smart watch application with vital sign monitoring and checklist reminders. In *AMIA Annual Symposium Proceedings*, volume 2015, page 314. American Medical Informatics Association, 2015.

[16] P. Barnaghi, P. Cousin, P. Maló, M. Serrano, and C. Viho. Simpler iot word (s) of tomorrow, more interoperability challenges to cope today. *River publishers series in communications*, page 277, 2013.

[17] P. Barnaghi, W. Wang, C. Henson, and K. Taylor. Semantics for the internet of things: early progress and back to the future. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 8(1):1–21, 2012.

[18] D. Beckett, T. Berners-Lee, and E. Prudâhommeaux. Turtle-terse rdf triple language. *W3C Team Submission*, 14(7), 2008.

[19] D. Beckett and B. McBride. Rdf/xml syntax specification (revised). *W3C recommendation*, 10(2.3), 2004.

[20] A. Berglund, S. Boag, D. Chamberlin, M. F. Fernández, M. Kay, J. Robie, and J. Siméon. Xml path language (xpath). *World Wide Web Consortium (W3C)*, 2003.

[21] T. Berners-Lee. Design issues: Linked data (2006). *URL http://www.w3.org/DesignIssues/LinkedData.html*, 2011.

[22] T. Berners-Lee et al. Semantic web road map, 1998.

[23] T. Berners-Lee, J. Hendler, O. Lassila, et al. The semantic web. *Scientific american*, 284(5):28–37, 2001.

[24] M. Birbeck and S. McCarron. Curie syntax 1.0–a syntax for expressing compact uris. w3c recommendation, 2009.

[25] P. V. Biron, A. Malhotra, et al. Xml schema part 2: Datatypes.

[26] C. Bizer, T. Heath, and T. Berners-Lee. Linked data-the story so far. *Semantic services, interoperability and web applications: emerging concepts*, pages 205–227, 2009.

[27] F. Bonomi, R. Milito, J. Zhu, and 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*, pages 13–16. ACM, 2012.

[28] A. Brajdic and R. Harle. Walk detection and step counting on unconstrained smartphones. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 225–234. ACM, 2013.

[29] T. Bray, J. Paoli, C. M. Sperberg-McQueen, E. Maler, and F. Yergeau. Extensible markup language (xml). *World Wide Web Journal*, 2(4):27–66, 1997.

[30] K. Breitman, M. A. Casanova, and W. Truszkowski. *Semantic web: concepts, technologies and applications.* Springer Science & Business Media, 2007.

[31] D. Brickley and R. V. Guha. Rdf vocabulary description language 1.0: Rdf schema. 2004.

[32] D. Brickley and L. Miller. Foaf vocabulary specification 0.99, namespace document 14 january 2014-paddington edition. 2014. http://xmlns.com/foaf/spec.

[33] G. Carothers and A. Seaborne. Rdf 1.1 n-triples. w3c recommendation. *World Wide Web Consortium, February*, 2014.

[34] D. Cerri and A. Fuggetta. Open standards, open formats, and open source. *Journal of systems and software*, 80(11):1930–1937, 2007.

[35] N.-C. Chen, K.-C. Wang, and H.-H. Chu. Listen-to-nose: a low-cost system to record nasal symptoms in daily life. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 590–591. ACM, 2012.

[36] S. Chen, H. Xu, D. Liu, B. Hu, and H. Wang. A vision of iot: Applications, challenges, and opportunities with china perspective. *IEEE Internet of Things journal*, 1(4):349–359, 2014.

[37] K.-H. Cheung, E. Prudâhommeaux, Y. Wang, and S. Stephens. Semantic web for health care and life sciences: a review of the state of the art, 2009.

[38] J. Clark. Xsl transformations (xslt). w3c recommendation, nov. 1999, 1998.

[39] R. J. Cole, D. F. Kripke, W. Gruen, D. J. Mullaney, and J. C. Gillin. Automatic sleep/wake identification from wrist activity. *Sleep*, 15(5):461–469, 1992.

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

[41] G. O. Consortium et al. The gene ontology (go) database and informatics resource. *Nucleic acids research*, 32(suppl 1):D258–D261, 2004.

[42] U. Consortium et al. *The Unicode Standard, Version 2.0.* Addison-Wesley Longman Publishing Co., Inc., 1997.

[43] W. Consortium et al. Xquery 1.0: An xml query language. *Version*, 1:W3C, 2007.

[44] O. Corcho, M. Fernández-López, and A. Gómez-Pérez. Methodologies, tools and languages for building ontologies. where is their meeting point? *Data & knowledge engineering*, 46(1):41–64, 2003.

[45] S. Cox and C. Little. Time ontology in owl. *https://www.w3.org/TR/owl-time/*, page 34, 2017.

[46] D. Crockford. The application/json media type for javascript object notation (json). 2006.

[47] B. Cuenca-Grau. Owl 1.1 web ontology language tractable fragments, 2007.

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

[49] M. C. Daconta, L. J. Obrst, and K. T. Smith. *The Semantic Web: a guide to the future of XML, Web services, and knowledge management.* John Wiley & Sons, 2003.

[50] S. Das, S. Sundara, and R. Cyganiak. R2rml: Rdb to rdf mapping language. w3c recommendation 27 september 2012. *Cambridge, MA: World Wide Web Consortium (W3C) (www.w3.org/TR/r2rml)*, 2012.

[51] A. K. Dey, K. Wac, D. Ferreira, K. Tassini, J.-H. Hong, and J. Ramos. Getting closer: an empirical investigation of the proximity of user to their smart phones. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 163–172. ACM, 2011.

[52] V. Dimitrieski, G. Petrović, A. Kovačević, I. Luković, and H. Fujita. A survey on ontologies and ontology alignment approaches in healthcare. In *International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, pages 373–385. Springer, 2016.

[53] A. Dimou, M. V. Sande, P. Colpaert, E. Mannens, and R. Van de Walle. Extending r2rml to a source-independent mapping language for rdf. In *Proceedings of the 2013th International Conference on Posters & Demonstrations Track-Volume 1035*, pages 237–240. CEUR-WS. org, 2013.

[54] A. Dimou, M. Vander Sande, P. Colpaert, R. Verborgh, E. Mannens, and R. Van de Walle. Rml: A generic language for integrated rdf mappings of heterogeneous data. In *LDOW*, 2014.

[55] R. Domina. Fascicolo sanitario elettronico da progetto a realtà: La situazione in italia (Electronic health record from project to reality: Current italian situation). *Available at SSRN: https://ssrn.com/abstract=2915696*, 2016.

[56] M. C. Domingo. An overview of the internet of things for people with disabilities. *Journal of Network and Computer Applications*, 35(2):584–596, 2012.

[57] F. M. Donini, M. Lenzerini, D. Nardi, and A. Schaerf. Reasoning in description logics. *Principles of knowledge representation*, 1:191–236, 1996.

[58] E. R. Dorsey, M. V. McConnell, S. Y. Shaw, A. D. Trister, S. H. Friend, et al. The use of smartphones for health research. *Academic Medicine*, 92(2):157–160, 2017.

[59] C. Doukas and I. Maglogiannis. Bringing iot and cloud computing towards pervasive healthcare. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012 Sixth International Conference on*, pages 922–926. IEEE, 2012.

[60] S. Duquennoy, G. Grimaud, and J.-J. Vandewalle. Smews: Smart and mobile embedded web server. In *Complex, Intelligent and Software Intensive Systems, 2009. CISIS'09. International Conference on*, pages 571–576. IEEE, 2009.

[61] O. C. Emine Sezer, Okan Bursa and M. O. Unalir. Semantic web technologies for iot-based health care information systems. In *SEMAPRO 2016 : The Tenth International Conference on Advances in Semantic Processing*, pages 45–48, 2016.

[62] D. Evans. The internet of things: How the next evolution of the internet is changing everything, 2011. cisco.com/c/dam/en_us/about/ac79/docs/innov/IoT_IBSG_0411FINAL.pdf.

[63] G. Eysenbach. The semantic web and healthcare consumers: a new challenge and opportunity on the horizon? *International Journal of Healthcare Technology and Management*, 5(3-5):194–212, 2003.

[64] L. Feigenbaum. Cambridge semantics position. In *W3C Workshop on RDF Next Steps, Stanford, Palo Alto, CA, USA*, 2010.

[65] D. Fensel, F. Van Harmelen, I. Horrocks, D. L. McGuinness, and P. F. Patel-Schneider. Oil: An ontology infrastructure for the semantic web. *IEEE intelligent systems*, 16(2):38–45, 2001.

[66] F. Fernandez and G. C. Pallis. Opportunities and challenges of the internet of things for healthcare: Systems engineering perspective. In *Wireless Mobile Communication and Healthcare (Mobihealth), 2014 EAI 4th International Conference on*, pages 263–266. IEEE, 2014.

[67] J. M. Ferranti, R. C. Musser, K. Kawamoto, and W. E. Hammond. The clinical document architecture and the continuity of care record: a critical analysis. *Journal of the American Medical Informatics Association*, 13(3):245–252, 2006.

[68] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager, et al. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79, 2010.

[69] F. Ganz, D. Puschmann, P. Barnaghi, and F. Carrez. A practical evaluation of information processing and abstraction techniques for the internet of things. *IEEE Internet of Things journal*, 2(4):340–354, 2015.

[70] M. Ganzha, M. Paprzycki, W. Pawłowski, P. Szmeja, and K. Wasielewska. Semantic interoperability in the internet of things: an overview from the inter-iot perspective. *Journal of Network and Computer Applications*, 81:111–124, 2017.

[71] G. Garkoti, S. K. Peddoju, and R. Balasubramanian. Detection of insider attacks in cloud based e-healthcare environment. In *Information Technology (ICIT), 2014 International Conference on*, pages 195–200. IEEE, 2014.

[72] V. Gay and P. Leijdekkers. Bringing health and fitness data together for connected health care: mobile apps as enablers of interoperability. *Journal of medical Internet research*, 17(11), 2015.

[73] S. Goessner. Jsonpath (2007). *URL http://goessner.net/articles/JsonPath.*

[74] C. Golbreich, E. K. Wallace, and P. F. Patel-Schneider. Owl 2 web ontology language new features and rationale. *W3C working draft, W3C (June 2009) http://www. w3. org/TR/2009/WD-owl2-new-features-20090611*, 2009.

[75] C. Golbreich, S. Zhang, and O. Bodenreider. The foundational model of anatomy in owl: Experience and perspectives. *Web Semantics: Science, Services and Agents on the World Wide Web*, 4(3):181–195, 2006.

[76] J. Gold. Fda regulators face daunting task as health apps multiply. *USA Today*, 2012.

[77] A. Gomez-Perez, M. Fernández-López, and O. Corcho. *Ontological Engineering: with examples from the areas of Knowledge Management, e-Commerce and the Semantic Web.* Springer Science & Business Media, 2006.

[78] B. N. Grosof, I. Horrocks, R. Volz, and S. Decker. Description logic programs: combining logic programs with description logic. In *Proceedings of the 12th international conference on World Wide Web*, pages 48–57. ACM, 2003.

[79] T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993.

[80] N. Guarino et al. Formal ontology and information systems. In *Proceedings of FOIS*, volume 98, pages 81–97, 1998.

[81] D. Guinard and V. Trifa. Towards the web of things: Web mashups for embedded devices. In *Workshop on Mashups, Enterprise Mashups and Lightweight Composition on the Web (MEM 2009), in proceedings of WWW (International World Wide Web Conferences), Madrid, Spain*, volume 15, 2009.

[82] A. Gyrard, C. Bonnet, and K. Boudaoud. Demo paper: Helping iot application developers with sensor-based linked open rules. In *TC/SSN@ ISWC*, pages 105–108, 2014.

[83] A. Gyrard, P. Patel, A. P. Sheth, and M. Serrano. Building the web of knowledge with smart iot applications. *IEEE Intelligent Systems*, 31(5):83, 2016.

[84] A. Gyrard and M. Serrano. A unified semantic engine for internet of things and smart cities: From sensor data to end-users applications. In *Data Science and Data Intensive Systems (DSDIS), 2015 IEEE International Conference on*, pages 718–725. IEEE, 2015.

[85] S. Handschuh and S. Staab. Annotation for the semantic web, frontiers in artificial intelligence and applications, vol. 96, 2003.

[86] J. Hendler and D. L. McGuinness. The darpa agent markup language. *IEEE Intelligent systems*, 15(6):67–73, 2000.

[87] C. Henson, A. Sheth, and K. Thirunarayan. Semantic perception: Converting sensory observations to abstractions. *IEEE Internet Computing*, 16(2):26–34, 2012.

[88] P. Heyvaert, A. Dimou, R. Verborgh, E. Mannens, and R. Van de Walle. Semantically annotating ceur-ws workshop proceedings with rml. In *Semantic Web Evaluation Challenge*, pages 165–176. Springer, 2015.

[89] P. Hitzler, M. Krötzsch, B. Parsia, P. F. Patel-Schneider, and S. Rudolph. Owl 2 web ontology language primer. *W3C recommendation*, 27(1):123, 2009.

[90] R. Hodgson and P. J. Keller. Qudt-quantities, units, dimensions and data types in owl and xml. *Online (November 2017) http://www.qudt.org*, page 34, 2011.

[91] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof, M. Dean, et al. Swrl: A semantic web rule language combining owl and ruleml. *W3C Member submission*, 21:79, 2004.

[92] I. Horrocks, P. F. Patel-Schneider, and F. Van Harmelen. Reviewing the design of daml+ oil: An ontology language for the semantic web. *AAAI/IAAI*, 2002:792–797, 2002.

[93] J. Hurwitz, M. Kaufman, and A. Bowles. *Cognitive computing and big data analytics*. John Wiley & Sons, 2015.

[94] I. Iakovidis. Towards personal health record: current situation, obstacles and trends in implementation of electronic healthcare record in europe. *International journal of medical informatics*, 52(1):105–115, 1998.

[95] D. Isern, D. Sánchez, and A. Moreno. Ontology-driven execution of clinical guidelines. *Computer methods and programs in biomedicine*, 107(2):122–139, 2012.

[96] S. R. Islam, D. Kwak, M. H. Kabir, M. Hossain, and K.-S. Kwak. The internet of things for health care: a comprehensive survey. *IEEE Access*, 3:678–708, 2015.

[97] R. S. Istepanian, S. Hu, N. Y. Philip, and A. Sungoor. The potential of internet of m-health things "m-iot" for non-invasive glucose level sensing. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 5264–5266. IEEE, 2011.

[98] M. Janssen, Y. Charalabidis, and A. Zuiderwijk. Benefits, adoption barriers and myths of open data and open government. *Information systems management*, 29(4):258–268, 2012.

[99] A. J. Jara, A. C. Olivieri, Y. Bocchi, M. Jung, W. Kastner, and A. F. Skarmeta. Semantic web of things: an analysis of the application semantics for the iot moving towards the iot convergence. *International Journal of Web and Grid Services*, 10(2-3):244–272, 2014.

[100] J. Jardine, J. Fisher, and B. Carrick. Apple's researchkit: smart data collection for the smartphone era?, 2015.

[101] R. Jasper, M. Uschold, et al. A framework for understanding and classifying ontology applications. In *Proceedings 12th Int. Workshop*

on Knowledge Acquisition, Modelling, and Management KAW, volume 99, pages 16–21, 1999.

[102] A. Jena. A free and open source java framework for building semantic web and linked data applications. *Available online:http://jena.apache.org*, 2015.

[103] Z. Jian, W. Zhanli, and M. Zhuang. Temperature measurement system and method based on home gateway. *Chinese Patent*, 102(811):185, 2012.

[104] M. R. Kamdar and M. J. Wu. Prism: A data-driven platform for monitoring mental health. In *PSB*, pages 333–344, 2016.

[105] K. Kawamoto, T. Tanaka, and H. Kuriyama. Your activity tracker knows when you quit smoking. In *Proceedings of the 2014 ACM international symposium on wearable computers*, pages 107–110. ACM, 2014.

[106] J. Kelly. Computing, cognition and the future of knowing. *Whitepaper, IBM Reseach*, 2015.

[107] H. H. Kim, S. Y. Lee, S. Y. Baik, and J. H. Kim. Mello: Medical lifelog ontology for data terms from self-tracking and lifelog devices. *International journal of medical informatics*, 84(12):1099–1110, 2015.

[108] K. Kim and E. Nahm. Benefits of and barriers to the use of personal health records (phr) for health management among adults. *Online Journal of Nursing Informatics OJNI*, 16(3):1–9, 2012.

[109] A. Kiryakov, B. Popov, I. Terziev, D. Manov, and D. Ognyanoff. Semantic annotation, indexing, and retrieval. *Web Semantics: Science, Services and Agents on the World Wide Web*, 2(1):49–79, 2004.

[110] P. Klasnja and W. Pratt. Healthcare in the pocket: mapping the space of mobile-phone health interventions. *Journal of biomedical informatics*, 45(1):184–198, 2012.

[111] N. Konstantinou and D.-E. Spanos. *Materializing the Web of Linked Data*. Springer, 2015.

[112] C. E. Koop, R. Mosher, L. Kun, J. Geiling, E. Grigg, S. Long, C. Macedonia, R. C. Merrell, R. Satava, and J. M. Rosen. Future delivery of health care: Cybercare. *IEEE Engineering in Medicine and Biology Magazine*, 27(6), 2008.

[113] L. Lacy, G. Aviles, K. Fraser, W. Gerber, A. M. Mulvehill, and R. Gaskill. Experiences using owl in military applications. In *OWLED*, volume 188, 2005.

[114] G. Lakoff. *Women, fire, and dangerous things*. University of Chicago press, 2008.

[115] E. C. Larson, M. Goel, G. Boriello, S. Heltshe, M. Rosenfeld, and S. N. Patel. Spirosmart: using a microphone to measure lung function on a mobile phone. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 280–289. ACM, 2012.

[116] D. Le-Phuoc and M. Hauswirth. Linked open data in sensor data mashups. In *Proceedings of the 2nd International Conference on Semantic Sensor Networks-Volume 522*, pages 1–16. CEUR-WS. org, 2009.

[117] J. Lee, B. A. Reyes, D. D. McManus, O. Maitas, and K. H. Chon. Atrial fibrillation detection using an iphone 4s. *IEEE Transactions on Biomedical Engineering*, 60(1):203–206, 2013.

[118] M. Lefrançois, A. Zimmermann, and N. Bakerally. A sparql extension for generating rdf from heterogeneous formats. In *European Semantic Web Conference*, pages 35–50. Springer, 2017.

[119] R. Lenz, M. Beyer, and K. A. Kuhn. Semantic integration in healthcare networks. *International journal of medical informatics*, 76(2):201–207, 2007.

[120] N. Lopes, A. Zimmermann, A. Hogan, G. Lukácsy, A. Polleres, U. Straccia, and S. Decker. Rdf needs annotations. In *W3C Workshop on RDF Next Steps, Stanford, Palo Alto, CA, USA*, 2010.

[121] W. Ludwig, K.-H. Wolf, C. Duwenkamp, N. Gusew, N. Hellrung, M. Marschollek, M. Wagner, and R. Haux. Health-enabling technologies for the elderly–an overview of services based on a literature

review. *Computer methods and programs in biomedicine*, 106(2):70–78, 2012.

[122] Y.-J. Ma, Y. Zhang, O. M. Dung, R. Li, and D.-Q. Zhang. Health internet of things: recent applications and outlook. *Journal of Internet Technology*, 16(2):351–362, 2015.

[123] F. Manola, E. Miller, B. McBride, et al. Rdf primer. *W3C recommendation*, 10(1-107):6, 2004.

[124] J. Manyika, M. Chui, J. Bughin, R. Dobbs, P. Bisson, and A. Marrs. Disruptive technologies: Advances that will transform life, business, and the global economy (vol. 12): Mckinsey global institute san francisco, ca, 2013.

[125] L. Masinter, T. Berners-Lee, and R. T. Fielding. Uniform resource identifier (uri): Generic syntax. 2005.

[126] M. Mauro. La nuova cultura della sanità dematerializzata. *Recenti Progressi in Medicina*, 105(11):408, 2014.

[127] D. L. McGuinness. *Ontologies come of age.* Mit Press, 2005.

[128] D. L. McGuinness, R. Fikes, J. Hendler, and L. A. Stein. Daml+ oil: an ontology language for the semantic web. *IEEE Intelligent Systems*, 17(5):72–80, 2002.

[129] D. L. McGuinness, F. Van Harmelen, et al. Owl web ontology language overview. *W3C recommendation*, 10(10):2004, 2004.

[130] R. V. Milani and C. J. Lavie. Health care 2020: reengineering health care delivery to combat chronic disease. *The American journal of medicine*, 128(4):337–343, 2015.

[131] E. Miller. An introduction to the resource description framework. *Bulletin of the Association for Information Science and Technology*, 25(1):15–19, 1998.

[132] G. Miller. Wordnet: An on-line lexical database. *International journal of lexicography*, 3(4):235–312, 1990.

[133] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[134] N. S. B. Miyoshi, A. A. P. Ferreira, and J. C. Felipe. Ontology-based approach to achieve semantic interoperability on exchanging and integrating information about the patient clinical evolution. In *Computer-Based Medical Systems, 2009. CBMS 2009. 22nd IEEE International Symposium on*, pages 1–6. IEEE, 2009.

[135] A. S. M. Mosa, I. Yoo, and L. Sheets. A systematic review of healthcare applications for smartphones. *BMC medical informatics and decision making*, 12(1):67, 2012.

[136] B. Motik, P. F. Patel-Schneider, and B. C. Grau. Owl 2 web ontology language direct semantics. *W3C recommendation*, 27, 2009.

[137] N. Z. Naqvib, A. Kumar, A. Chauhan, and K. Sahni. Step counting using smartphone-based accelerometer. *International Journal on Computer Science and Engineering*, 4(5):675, 2012.

[138] T. H. Nelson. Complex information processing: a file structure for the complex, the changing and the indeterminate. In *Proceedings of the 1965 20th national conference*, pages 84–100. ACM, 1965.

[139] N. Noy, A. Rector, P. Hayes, and C. Welty. Defining n-ary relations on the semantic web. *W3C working group note*, 12(4), 2006.

[140] N. F. Noy, M. Sintek, S. Decker, M. Crubézy, R. W. Fergerson, and M. A. Musen. Creating semantic web contents with protege-2000. *IEEE intelligent systems*, 16(2):60–71, 2001.

[141] Z. Pang. *Technologies and Architectures of the Internet-of-Things (IoT) for Health and Well-being*. PhD thesis, KTH Royal Institute of Technology, 2013.

[142] J. A. Paradiso and T. Starner. Energy scavenging for mobile and wireless electronics. *IEEE Pervasive computing*, 4(1):18–27, 2005.

[143] A. Park, H. Chang, and K. J. Lee. Action research on development and application of internet of things services in hospital. *Healthcare informatics research*, 23(1):25–34, 2017.

[144] P. Pediaditis, G. Flouris, I. Fundulaki, and V. Christophides. On explicit provenance management in rdf/s graphs. In *Workshop on the Theory and Practice of Provenance*, 2009.

[145] P. Pelegris, K. Banitsas, T. Orbach, and K. Marias. A novel method to detect heart beat rate using a mobile phone. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 5488–5491. IEEE, 2010.

[146] D. Pfisterer, K. Romer, D. Bimschas, O. Kleine, R. Mietz, C. Truong, H. Hasemann, A. Kröller, M. Pagel, M. Hauswirth, et al. Spitfire: toward a semantic web of things. *IEEE Communications Magazine*, 49(11):40–48, 2011.

[147] I. Plaza, L. MartíN, S. Martin, and C. Medrano. Mobile applications in an aging society: Status and trends. *Journal of Systems and Software*, 84(11):1977–1988, 2011.

[148] M.-Z. Poh, D. J. McDuff, and R. W. Picard. Advancements in noncontact, multiparameter physiological measurements using a webcam. *IEEE transactions on biomedical engineering*, 58(1):7–11, 2011.

[149] E. Prud, A. Seaborne, et al. Sparql query language for rdf. 2006.

[150] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman. Activity recognition from accelerometer data. In *Aaai*, volume 5, pages 1541–1546, 2005.

[151] R. Rawassizadeh, B. A. Price, and M. Petre. Wearables: Has the age of smartwatches finally arrived? *Communications of the ACM*, 58(1):45–47, 2015.

[152] B. Reeder and A. David. Health at hand: a systematic review of smart watch uses for health and wellness. *Journal of biomedical informatics*, 63:269–276, 2016.

[153] A. Rhayem, M. B. A. Mhiri, M. B. Salah, and F. Gargouri. Ontology-based system for patient monitoring with connected objects. *Procedia Computer Science*, 112:683–692, 2017.

[154] F. J. Riggins and S. F. Wamba. Research directions on the adoption, usage, and impact of the internet of things through the use of big data analytics. In *System Sciences (HICSS), 2015 48th Hawaii International Conference on*, pages 1531–1540. IEEE, 2015.

[155] J. Rowley. The wisdom hierarchy: representations of the dikw hierarchy. *Journal of information science*, 33(2):163–180, 2007.

[156] M. Ruiz, J. García, and B. Fernández. Body temperature and its importance as a vital constant. *Revista de enfermeria (Barcelona, Spain)*, 32(9):44–52, 2009.

[157] G. Santucci et al. From internet of data to internet of things. In *International Conference on Future Trends of the Internet*, volume 28, 2009.

[158] R. Shearer, B. Motik, and I. Horrocks. Hermit: A highly-efficient owl reasoner. In *OWLED*, volume 432, page 91, 2008.

[159] A. Sheth. Internet of things to smart iot through semantic, cognitive, and perceptual computing. *IEEE Intelligent Systems*, 31(2):108–112, 2016.

[160] A. Sheth, U. Jaimini, K. Thirunarayan, and T. Banerjee. Augmented personalized health: How smart data with iots and ai is about to change healthcare. In *Research and Technologies for Society and Industry (RTSI), 2017 IEEE 3rd International Forum on*, pages 1–6. IEEE, 2017.

[161] A. Sidhu, T. S. Dillon, E. Chang, and B. Sidhu. Protein ontology development using owl. In *Proceedings of the 2005 Workshop on OWL: Experiences and Directions (OWLED'05)*. CEUR Workshop Proceedings, 2005.

[162] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz. Pellet: A practical owl-dl reasoner. *Web Semantics: science, services and agents on the World Wide Web*, 5(2):51–53, 2007.

[163] C. Snomed. Systematized nomenclature of medicine-clinical terms. *International Health Terminology Standards Development Organisation*, 2011.

[164] D. Soergel, B. Lauser, A. Liang, F. Fisseha, J. Keizer, and S. Katz. Reengineering thesauri for new applications: the agrovoc example. *Journal of digital information*, 4(4), 2006.

[165] J. F. Sowa et al. *Knowledge representation: logical, philosophical, and computational foundations*, volume 13. MIT Press, 2000.

[166] D. M. Steinberg, D. F. Tate, G. G. Bennett, S. Ennett, C. Samuel-Hodge, and D. S. Ward. The efficacy of a daily self-weighing weight loss intervention using smart scales and e-mail. *Obesity*, 21(9):1789–1797, 2013.

[167] C. P. Stone. A glimpse at ehr implementation around the world: The lessons the us can learn. *Health Institute for E-Health Policy, May*, 2014.

[168] X. Su, H. Tong, and P. Ji. Activity recognition with smartphone sensors. *Tsinghua Science and Technology*, 19(3):235–249, 2014.

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

[170] H. J. ter Horst. Completeness, decidability and complexity of entailment for rdf schema and a semantic extension involving the owl vocabulary. *Web Semantics: Science, Services and Agents on the World Wide Web*, 3(2):79–115, 2005.

[171] E. J. Topol. The big medical data miss: challenges in establishing an open medical resource. *Nature Reviews Genetics*, 16(5):nrg3943, 2015.

[172] K. Tran, T. Le, and T. Dinh. A high-accuracy step counting algorithm for iphones using accelerometer. In *Signal Processing and Information Technology (ISSPIT), 2012 IEEE International Symposium on*, pages 000213–000217. IEEE, 2012.

[173] D. Tsarkov and I. Horrocks. Fact++ description logic reasoner: System description. *Automated reasoning*, pages 292–297, 2006.

[174] K. Tsubouchi, R. Kawajiri, and M. Shimosaka. Working-relationship detection from fitbit sensor data. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 115–118. ACM, 2013.

[175] M. Uschold and M. Gruninger. Ontologies: Principles, methods and applications. *The knowledge engineering review*, 11(2):93–136, 1996.

[176] A. J. J. Valera, M. A. Zamora, and A. F. Skarmeta. An architecture based on internet of things to support mobility and security in medical environments. In *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*, pages 1–5. IEEE, 2010.

[177] F. Van den Abeele, J. Hoebeke, I. Moerman, and P. Demeester. Integration of heterogeneous devices and communication models via the cloud in the constrained internet of things. *International Journal of Distributed Sensor Networks*, 11(10):683425, 2015.

[178] F. Vitali, A. Amoroso, M. Roccetti, and G. Marfia. Restful services for an innovative e-health infrastructure: a real case study. In *e-Health Networking, Applications and Services (Healthcom), 2014 IEEE 16th International Conference on*, pages 188–193. IEEE, 2014.

[179] T. Wadhawan, N. Situ, H. Rui, K. Lancaster, X. Yuan, and G. Zouridakis. Implementation of the 7-point checklist for melanoma detection on smart handheld devices. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 3180–3183. IEEE, 2011.

[180] L. Wei, Y. Heng, and W. Y. Lin. Things based wireless data transmission of blood glucose measuring instruments. *Chinese Patent*, 202(154):684, 2012.

[181] C. Welty, D. L. McGuinness, and M. K. Smith. Owl web ontology language guide. *W3C recommendation, W3C (February 2004) http://www. w3. org/TR/2004/REC-owl-guide-20040210*, 2004.

[182] P. L. Whetzel, N. F. Noy, N. H. Shah, P. R. Alexander, C. Nyulas, T. Tudorache, and M. A. Musen. Bioportal: enhanced functionality via new web services from the national center for biomedical ontology

to access and use ontologies in software applications. *Nucleic acids research*, 39(suppl_2):W541–W545, 2011.

[183] G. Witmer. Dictionary of philosophy of mind- ontology. *Retrieved May*, 11:2004, 2004.

[184] L. Wood, A. Le Hors, V. Apparao, S. Byrne, M. Champion, S. Isaacs, I. Jacobs, G. Nicol, J. Robie, R. Sutor, et al. Document object model (dom) level 1 specification. *W3C Recommendation*, 1, 1998.

[185] Q. Wu, G. Ding, Y. Xu, S. Feng, Z. Du, J. Wang, and K. Long. Cognitive internet of things: a new paradigm beyond connection. *IEEE Internet of Things Journal*, 1(2):129–143, 2014.

[186] D. Zeng, S. Guo, and Z. Cheng. The web of things: A survey. *Journal of Communications*, 6(6):424–438, 2011.

[187] X. Zenuni, B. Raufi, F. Ismaili, and J. Ajdari. State of the art of semantic web for healthcare. *Procedia-Social and Behavioral Sciences*, 195:1990–1998, 2015.