Communauté UNIVERSITÉ Grenoble Alpes

THÈSE

Pour obtenir le grade de

DOCTEUR DE la Communauté UNIVERSITÉ GRENOBLE ALPES

Spécialité : **Informatique**

Arrêté ministériel : 27 Janvier 2017

Présentée par

Thiago Moreira da Costa

Thèse dirigée par **Hervé Martin**
et codirigée par **Nazim Agoulmine**

préparée au sein **LIG - Laboratoire d'Informatique de Grenoble**
et de **l'EDMSTII - l'Ecole Doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

OPP-IoT

An ontology-based privacy preservation approach for the Internet of Things

Thèse soutenue publiquement le **27 Janvier 2017**,
devant le jury composé de :

M, Didier DONZES
Docteur, Pr. à l'Université Grenoble Alpes, Président

MME, Karine ZEITOUNI
Docteur, Pr. à l'Université de Versailles-Saint-Quentin-en-Yvelines, Rapporteur

MME, Maryline LAURENT
Docteur, Pr. à l'Université Telecom SudParis, Rapporteur

M, Reinaldo BRAGA
Docteur, Pr. à l'Institut Fédéral du Ceará, Examinateur

To my beloved parents...

ABSTRACT

The spread of pervasive computing through the Internet of Things (IoT) represents a challenge for privacy preservation. Privacy threats are directly related to the capacity of the IoT sensing to track individuals in almost every situation of their lives. Allied to that, data mining techniques have evolved and been used to extract a myriad of personal information from sensor data stream. This trust model relies on the trustworthiness of the data consumer who should infer only intended information. However, this model exposes personal information to privacy adversary. In order to provide a privacy preservation for the IoT, we propose a privacy-aware virtual sensor model that enforces privacy policy in the IoT sensing. This mechanism intermediates physical sensors and data consumers. As a consequence, we are able to optimize the use of privacy preserving techniques by applying them selectively according to virtual sensor inference intentions, while preventing malicious virtual sensors to execute or get direct access to raw sensor data. In addition, we propose an ontology to classify personal information based on the Behavior Computing, facilitating privacy policy definition and information classification based on the behavioral contexts.

RÉSUMÉ

La vulgarisation de l'informatique omniprésente à travers l'internet des objets (IdO) représente un défi pour la préservation de la vie privée et la confidentialité des individus. Les menaces contre la confidentialité sont directement liées à la capacité de détection de capteur dans l'IdO en suivant les individus dans presque toutes les situations de leur vie. Alliée à cela, les techniques d'exploration de données ont évolué et ont été utilisées pour extraire une multitude d'informations personnelles à partir de données du flux des données des capteurs. Ce modèle de confiance repose sur la fiabilité du consommateur de données pour extraire uniquement des informations accordées. Cependant, ce modèle permet l'exposition d'informations personnelles à des adversaires de la vie privée. Afin de fournir un mécanisme pour préserver la confidentialité dans l'IdO, nous proposons un modèle de capteur virtuel qui renforce une politique de confidentialité dans le flux des données des capteurs. Ce mécanisme intermédiaire se met en place entre les capteurs physiques et les consommateurs de données. En conséquence, nous sommes en mesure d'optimiser l'utilisation des techniques de preservation de confidentialité, telles qu'anonymisation, en les appliquant de manière sélective selon les intentions d'inférence des capteurs virtuelles, tout en empêchant les capteurs virtuels malveillants d'exécuter ou d'obtenir un accès direct aux données brutes des capteurs physiques. En outre, nous proposons une ontologie pour classer les informations personnelles basées sur la science du comportement (Behavior Computing), ce qui facilite la définition de la politique de confidentialité et à la classification de l'information en fonction des contextes comportementaux.

PUBLICATIONS

— **Thiago Moreira da Costa, Hervé Martin, and Nazim Agoulmine**. "Privacy-Aware personal Information Discovery model based on the cloud." Network Operations and Management Symposium (LANOMS), 2015 Latin American. IEEE, 2015. Alagoas, Brazil.

— **Emmanuel Coutinho, Macerlo Santos, Stenio Fernandes, José Neuman de Souza, Thiago Moreira da Costa, Elie Rachkidi, Nazim Agoulmine, and Javier Baliosian**, "Research Opportunities in an Intercloud Environment Using MOSt in SLA4CLOUD Project," in International Workshop on ADVANCEs in ICT Infrastructures and Services. Fortaleza, Brazil.

— **Thiago Moreira da Costa, Elie Rachkidi and Leonardo M. Gardini, César Moura, Luiz O. M. Andrade and Mauro Oliveira** "An Architecture for LARIISA: An Intelligent System for Decision Making and Service Provision in Health Care," 4th International Workshop on ADVANCEs in ICT Infrastructures and Services, 2015. Fortaleza, Brazil.

— **Thiago Moreira da Costa, Elie Rachkidi, Nazim Agoulmine, and Herve Martin**, 'An experiment on deploying a privacy-aware sensing as a service in the sensor cloud,' in IEEE ADVANCESs in ICT, 2017. Paris, France

ACKNOWLEDGMENTS

Foremost, I would like to express my sincere gratitude to my supervisor Prof. Hervé Martin and my advisor Prof. Nazim Agoulmine for the continuous support of my research, and for their enthusiasm, patience, and guidance. In particular, I would like to thank Prof. Hervé Martin for accepting me in the Ph.D. candidacy and being a rock to overcome obstacles that I found before and during this journey. I would like to thank Nazim Agoulmine, as well, for accepting the challenge of co-orientating my work, for inviting me to work with his team at the Université d'Évry, and for being my weekly contact.

I would also like to thank the rest of my thesis committee: Prof. Karine ZEITOUNI, prof. Maryline LAURENT, prof. Reinaldo BRAGA, and prof. Didier DONZES, for their insightful review comments, clarifying remarks, and highly relevant questions.

I thank CAPES and the MEC (the Brazilian Ministry of Education and Communication) for the financial support, the LIG (Laboratoire d'Informatique de Grenoble), STEAMER team, Paule-Annick, Philipe for having me and supporting my research. Thank you, Frank, Elie, and the IBISC team for having me at the laboratory, and contributing to the quality of my work. In particular, my labmate Elie who joined sleepless nights to meet several 12PM ESTs (UTC/GMT - 5 hours).

My sincere thanks also go to my family who supported me with all possible manners, giving me the strength to step forward when my mind was tired and when troubled water was under the bridge. I thank my dad, mom, brother, and aunt Janete. My sincere gratitude to Simon, for being my family in Paris and supporting me in every situation; to Humberto for all support, conversation, and fraternity, and limitless affinity. Without all your love, I am not sure if I would be able to accomplish this alone. I want to thank Rafael for supporting me and believing in my potential since the beginning. My thanks also go to beloved friends who made life easier during these 3 years: Fernando, David, James, Rolando, Joseane, Fábio, Francóis, Marc-Olivier, Alexis, Ana, Peter, Pierre, and Nicolas, for each moment of conversation, nights out, trips, and dinners. Reinaldo and Carina, thank you both for believing in my potential since 2008! and pushing me to go that extra mile to achieve my dreams. And above all, I thank you, coffee!

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

3G Third generation of wireless mobile telecommunications

4G-LTE Forth generation of wireless mobile telecommunications

5G Next generation of wireless network

ABAC Attribute-based Access Control

ABox Assertion Components

ACM Access Control Model

ACPC Access Control Policy Condition

ACVS Access Control Virtual Sensor

API Application Programming Interface

BC Behavior Computing

BFO Basic Formal Ontology

CASTLE Continuously Anonymizing STreaming data via adaptive cLustEring

CEP Complex Event Processing

CQL Continuous Query Language

DAC Discretionary Access Control

DaaS Data as a Service

DBaaS Database as a Service

DIE Directive Information Entity

DIKW Data-Information-Knowledge-Wisdom

DL Description Logic

DMOP Data Mining OPtimiation Ontology

DnS Description & Situations Ontology

DOLCE Descriptive Ontology for Linguistic and Cognitive Engineering

DSM Data Stream Mining

DSMS Data Stream Management System

DUL DOLCE-DnS UltraLite

ENVO Environment Ontology

EXACT Experiment ACTions Ontology

Exposé Exposé Ontology

FOAF Friends Of A Friend Ontology

FTP File Transfer Protocol

GO Gene Ontology

GSN Global Sensor Network

GPS Global Positioning System

H2M Human to Machine

HTTP Hypertext Transfer Protocol

IaaS Infrastructure as a Service

IAO Information Artifact Ontology

ICE Information Content Entity

ICO Informed Consent Ontology

ICT Information and Communications Technology

IoT Internet of Things

IPMaaS Identity and Policy Management as a Service

IPv4 Internet Protocol version 4

IPv6 Internet Protocol version 6

IRI Internationalized Resource Identifier

IT Information Technology

KDD Knowledge Discovery in Databases

KDDM Knowledge Discovery and Data Mining

LABORS LAboratory Ontology for Robot Scientists

LSM Linked Stream Middleware

M2M Machine to Machine

MAC Mandatory Access Control

MF Mental Functioning Ontology

MIREOT Minimum Information to Reference an External Ontology Term

MM Meta-Mining

NBO Neuro Behavioral Ontology

NFC Near Field Communication

NIST National Institute of Standard and Technologies

OBI Ontology for Biomedical Investigation

OBO Open Biomedical Ontologies

OGC Open Geospatial Consortium

OGMS Ontology for General Medical Science

O&M Observations and Measurements Schema

OntoDT Ontology for Data type

OntoDM Ontology of Data Mining

OntoDM-KDD OntoDM for KDD

OntoFox ONTOlogy tool that Fetch Ontology terms and aXioms

OPIS Ontology for Personal Information on the Sensor Web

OS Operating System

OWL Ontology Web Language

OWLDL OWL Description Logic

OWLTime Time Ontology in OWL

PaaS Platform as a Service
PA-VSM Privacy-aware Virtual Sensor Model
PATO Phenotypic quality
PC Perception Computing
PEP Privacy Enforcement Point
PET Privacy-Enhancing Technology
PIL Personal Information Layer
PIMO Personal Information Model Ontology
PIS Privacy-friendly Information System
PRSS Personal RDF Stream Sample
PPC Privacy Policy Condition
PPDMT Privacy-Preserving Data Mining Technique
PPPC Privacy-Preserving Policy Condition
PPVS Privacy-Preserving Virtual Sensor
QoS Quality of Service
RBAC Role-based Access Control
RDBMS Relational Database Management System
RDF Resource Description Framework
RDFS RDF Schema
RFID Radio-Frequency IDentification
OBO RO OBO Relational Ontology
RuleML Rule Markup Language
SAaaS Sensing and Actuation as a Service
SaaS Software as a Service
SABRE Sensitive Attribute Bucketization and REdistribution framework
SEaaS Sensor as a Service
SensorML Sensor Model Language
SLA Service Level Agreement
SOA Service-Oriented Architecture
SP Semantic Perception
SPARQL SPARQL Protocol and RDF Query Language
SPARQL-DL SPARQL-DL Language
SPL Semantic Perception Layer
SPP Semantic Perception Process
S^2aaS Sensing as a Service
SSN Semantic Sensor Network
SSN-O Semantic Sensor Network Ontology
SSO Stimulus-Sensor-Observation
SSR Semantic Stream Reasoning

SUID Standard Ubiquitous ID
SUMO Suggested Upper Merged Ontology
SWE Sensor Web Enablement
SWO Software Ontology
SWRL Semantic Web Rule Language
SWRLTO SWRL Temporal Ontology
TaaS Thing as a Service
TAO Temporal Abstractions Ontology
TBox Terminological Components
URI Uniform Resource Identifier
URL Uniform Resource Locator
VIVO-ISF Ontology for clinical research and base expertise
VSaaS Video Surveillance as a Service
VSN Virtual Sensor Network
XACML eXtensive Access Control Markup Language
xOPIS eXtended-OPIS
XaaS Everything as a Service
xGSN eXtended Global Sensor Network
W3C World Wide Web Consortium
WEKA Waikato Environment for Knowledge Analysis
WLAN Wireless Local Area Network
WWW World Wide Web
XML Extensible Markup Language
PIM Personal Information Management

1 INTRODUCTION

1.1 CONTEXT

The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it". — Mark Weiser

Pervasive computing is spreading as sensors map the physical reality and its phenomena into digital traces, providing a fertile research area to explore statistical models that could be interpreted and correlated to a variety of personal information. The development of pervasive computing has incorporated the advances in sensing, communication, networking, web service, and information processing technologies, creating a new era of connected *things*, so called the Internet of Things (IoT) [1].

Along with the Cloud Computing, IoT boosted the sensing capacity of the pervasive computing by delivering scalable, virtualized and nearby (edge computing) resources. Besides that, the *pay-as-you-go* model offered by the Cloud Computing, which traditionally delivers services at the layers of infrastructure, platform, and software; promoted the development of a wide range of specialized services, such as sensing, networking, and specific software-based functionalities [2].

The gradual increment of the sensing omnipresence has, as a consequence, allowed observing individuals in numerous situations. Allied to that, the intensive usage of Knowledge Discovery and Data Mining (KDDM) techniques to interpret this data has leveraged the capacity to perceive and learn about individuals and their behaviors. Consequently, a new generation of interconnected smart applications, smart devices and smart actuators that interpret these observations – and respond to them – are emerging in different fields, such as social networks, e-Health, mobility, environmental monitoring, and smart cities [3]. This has raised concerns about privacy in the IoT both from the research community and the industry [4].

From a legal perspective, the IoT has contributed to a discussion about society, politics, and market, which has been reflected in recent changes to countries' privacy regulation and data protection laws that reinforce citizens' privacy and address data protection and ownership issues, such as the new European Union directives [5]. These regulations are being strategically negotiated between countries. Recently the European Commission and the United States reached an agreement on a new framework to permit cross-border transfers of personal data [6], pushing higher requirements and obligations on American *data consumers* [1] by requiring transparency, accountability, law and privacy mechanisms to ensure European citizens' rights.

1. Data consumers are those who are interested in the use and value of the personal content.[7]

From a personal perspective, individuals (*data owners* [2]) are highly concerned about their privacy [8]. However, their privacy risk perception [9] and privacy behavior [10] are paradoxical, since they continue consuming services that explicitly outbreak their privacy. This is partly explained by the choices in privacy strategies and mechanisms that privilege data processing in *data consumers'* side, which often mislead users to trust *data consumers* and to reveal more information than necessary by offering access to raw sensor data, such as geographic location, camera, microphone, or step counters from their smartphones. Therefore, trust and ethical issues related to the manipulation and discovery of personal information become more critical in the IoT given its pervasiveness.

In this new politic, economical and social era ignited by the IoT, we wonder what is the place of privacy and technology in this complex scenario.

1.2 PRIVACY

"The need for privacy is a socially created need. Without society there would be no need for privacy" — Barrington Moore

Privacy is a multidimensional concept associated with cultural, political, economical, social, and temporal aspects that cause *social friction* and individual harm. The consequences associated with privacy problems go beyond mental pain or distresses, such as reputational or dignitary injury, risking to damage physically, economically or politically a person [11]. These *social frictions* are inherent to life in society and have molded the conceptualization of privacy from the first scholar reference of *physical privacy* – "*the right to be left alone*" [12] – to the contemporary data protection regulation that covers *informational privacy*, such as "*the right to be forgotten*" [13]. In the exercise of their right to seclusion, individuals can freely behave and express themselves without the threat of censure, and thus exercising their moral right to autonomy [14]. This notion of privacy based on the seclusion is redefined by Nissenbaum [15] who argues that violation of privacy is a function of several variables, which includes situation (context), the roles of agents receiving information and their relationship to information subjects, how the information is shared by the individual and by the agents. The definition of privacy, therefore, is an evolving concept and should be understood in the light of social and behavioral contexts in different eras.

In the information age, individual's dignity and physical integrity are affected by opportunistic harmful activities that may take place during the information collection, processing, and dissemination. The morality system code of conduct and social judgment can be used as scrutiny of individual's behavior and decision where surveillance through continuous collection and inference of sensor data and personal information act to intimidate individual freedom. The exposure of private behaviors through information system should

2. Data owners are subject or producers of the personal content and who owns it.[7]

be carried more cautiously than previous privacy threats since the extraction of information through recorded and sensed behaviors are more detailed and stay indeterminately available for processing techniques often more efficient than human analysis capacities. The moral impact of privacy invasion extrapolates the ethical behavior and decisions as "*data privacy protection furthers still another sort of liberty–that of self-determination, expressed through the power to define oneself to the world in the way one wishes*" [16]. As argued by Foucault, this surveillance is a type of continuous oppressive presence of a multiple, automatic, and anonymous power that excesses indirectly force and violence and induces passivity in the surveilled [17]. This passivity in the technological information world, in particular in the IoT, therefore, is distilled with limited choices and privacy protection mechanisms. As a consequence, individuals are led with motivation to pursue their expressed interests based on these limited choices, following the path of least resistance and friction. Foucault's epistemological perspective could not be more contemporary, as individuals are led to accept privacy intrusion by continuous surveillance through the IoT, and where privacy mechanisms are offered conveniently by *data consumers* who claim to have ethics to protect and process personal data. This established *trust model* has been assumed mostly as the only choice for the dissemination and discovery of personal information through the IoT. However, privacy preservation in the IoT should be investigated and addressed from another perspective that privileges the interests of *data owners* instead of those of *data consumers*.

1.3 PERSONAL INFORMATION AND BEHAVIOR

Currently, the *Oxford Dictionary* defines *information* as a fact provided or learned about something or someone; or as something that is conveyed or represented by a particular arrangement or sequence of things. In information systems, *information* is defined as "*an assembly of data in a comprehensive form capable of communication and use*" [18]. In the remainder of this manuscript, we adopted Ackoff's vision of the Data-Information-Knowledge-Wisdom (DIKW) triangle, as depicted in Figure 1.1. In DIKW, each level is generated by the observation and generalization of the lower-level *informational entity*. All these information entities represent properties, objects, individuals, events, abstract concepts, and information entity itself in different abstraction levels.

Personal information is a commonly overloaded term referring both to an individual and information about oneself; as well as to information controlled and owned by someone [20]. In this thesis, *personal information* is defined as *all data, information, knowledge, and wisdom related to an individual and/or under her control*.

The sheer volume of data generated by the IoT sensing requires an adaptation in the perceptive of personal information for inference and privacy purposes. The pervasiveness and ubiquitousness of the IoT

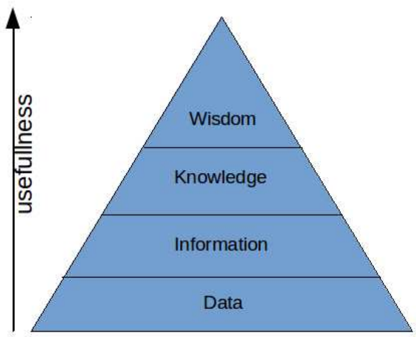

Figure 1.1 – The Data-Information-Knowledge-Wisdom triangle [19]

sensing cause a continuous observation and surveillance, mapping physical phenomena and behaviors to sensor data. Along with data collection, the data processing and inference aggregate semantics to the informational entity in each DIKW level.

Most privacy approaches for the IoT target access control and policy enforcement at the lowest informational layer (sensor data), while applications (*data consumers*) and individuals (*data owners*) tend to privilege the highest informational levels (wisdom, knowledge, and information). Moreover, the behavioral aspect of the IoT sensing for personal information is not considered, limiting the chances to contextualize inference and privacy policy conditions based on the individual's situation. We claim that the behavioral model found in the research domain of Behavioral Computing [21] can be applied to model personal information efficiently.

The Behavior Computing (BC) is a systematic way to structure and study behaviors (events) using a set of concepts, definitions, and tools to represent, explain and predict behaviors. For BC, *behavior* is defined as actions, operations or events conducted by agents within certain context and environment (virtual or physical ones), focusing on symbolic behaviors that represent these activities into a computational model. Thereby, it can be stated that BC intends to enrich the process of behavior pattern analysis, by modeling features and allowing in-depth analysis of behavior and its impacts.

Based on the premise that observations are made in behavioral temporal frames using sensors, it is possible to model different levels of abstraction of personal information, involved features around the concept of behavior and its temporal dimension. As a result, we are able to represent information about the individual (agent), the involved physical, mental and abstract features that participates in the observed behavior, and the behavior itself.

1.4 MOTIVATION AND PROBLEM STATEMENT

The research in privacy engineering points to requirements of design, strategy, and mechanisms to address privacy issues in systems [22]. In this manuscript, we focus on the privacy preservation for the IoT sensing with regard to its design, considering a *privacy-by-policy* strategy. More specifically, we address privacy issues associated with the Sensing as a Service (S^2aaS) [3] [23] model (*privacy-by-design*) in the IoT and the privacy preservation strategy on top of the S^2aaS implemented by a mechanism that enforces *privacy-by-policy*. In order to assess the IoT from a privacy engineering viewpoint, we need to understand its enabling technologies, architectures and the recent incorporation of Cloud Computing to IoT sensing services. Given its novelty and complexity, studies about privacy issues in the IoT are incipient and sparse, which hamper the development of approaches to the design privacy-sensitive IoT systems [24], and to determine requirements that needs to be met to evaluate the efficacy of the proposed privacy mechanisms. Motivated by these concerns, we state the following research questions:

— **Q1**: How to design a privacy model for the IoT sensing as a service S^2aaS to enable a more efficient approach to the design of privacy preserving IoT systems?

— **Q2**: How to design a *privacy-by-policy* mechanism in the S^2aaS that facilitates the definition of the privacy policy by end users based on intelligible personal information and its context?

— **Q3**: Are these privacy model and *privacy-by-policy* mechanism implementable in an IoT testbed platform?

Available approaches rely on Privacy-Enhancing Technologies (PETs) conventionally on the *data owners*' side (user sphere) or on the *data consumer*'s side (recipient sphere). Both paradigms pose challenges due to its trust model, resource limitation, and architecture. The user sphere may safely prevent private data to be released since it is completely under data owner's control. On the other hand, it may limit the development of more sophisticated privacy preserving mechanisms that demand computational power or penalize resource-constrained devices, which are commonly found in the IoT objects [4], by draining their resources.

Conversely, once data and personal information are released to the recipient sphere, privacy adversaries or malicious *data consumers* can exploit privacy breaches, extracting unintended personal information from the published data, even from anonymized datasets [25]. Besides that, the increasing number of portable devices that publish sensor data related to the participatory sensing of the IoT objects has been associated with dense, dynamic, location-aware and onerous to manage networks of device [26]. Therefore, server-centric solutions where personal information is published or kept on *data consumer*'s control present several privacy threats and architectural challenges, demanding an alternative model to minimize these issues.

In addition, few of these approaches have considered the cognitive bias that plays an important role in the definition of policies, which should reflect the *data owner*'s concerns in an intelligible manner. In general, sensor observations (and other lower level data that is ex-

3. The acronym S^2aaS is adopted in this manuscript with the only objective of differentiating it from the traditional SaaS acronym normally related to Software as a Service.

4. IoT objects correspond to physical devices connected to the IoT network.

changed among IoT entities [5]) demand technical knowledge to be understood and classified. By employing Semantic Web technology to overcome the barriers of knowledge representation and knowledge interpretation, some privacy preservation approaches have proposed Access Control Models (ACMs) based on ontologies, such as the SSN-O. However, these approaches do not provide an ontology capable of bridging the semantic sensor network world, that is currently used to annotate sensor data in the IoT, to a higher-level information, that would permit end-users to express their privacy concerns more accurately.

Allied to that, the use of KDDM techniques, that support the discovery of personal information, makes humanly impossible to classify sensor data in terms of potential malicious inference, due to the multitude of techniques and possible statistical correlation of their results to personal information. Privacy-Preserving Data Mining Techniques (PPDMTs) [6] are commonly used in these scenarios to degrade data utility in order to minimize the chance of extracting private information. However, the direct application of these techniques degrades data utility continuously, even when there is no private information to hide, consuming computational resources unnecessarily. Moreover, PPDMT approaches suffer from cyclic *re-identifications* [7] which restricts the usefulness of such techniques.

Many attempts to provide privacy mechanisms in the IoT sensing are proposed, but as we will explain, they consider neither modern privacy engineering principles in its design nor comprehensive *privacy-by-policy* mechanisms tailored for the IoT.

1.5 OBJECTIVES AND APPROACH

In this thesis, we aim at providing an approach for privacy preservation in the IoT through a privacy model and a *privacy-by-policy* mechanism for the S^{2}aaS. Since the system design plays a crucial role in privacy preservation, we aim at redefining a privacy model based on modern privacy engineering principles and the available IoT enabling technologies. In this direction, we envision a *privacy-aware Sensing as a Service (S^{2}aaS)* using the Cloud-IoT infrastructure that intermediates IoT objects (*data providers*) [8] and IoT applications (*data consumers*) with a Virtual Sensor Network (VSN), as depicted in Figure 1.2.

In order to provide this, we address the three research questions specified in the previous section by:

— Providing a privacy model for the IoT S^{2}aaS to intermediate IoT objects (data providers) and IoT applications (data consumers)

5. IoT entities refer to middleware, system, application, and object that exchange data in the IoT network.

6. The acronym stands for the abbreviation for *Privacy-Preserving Data Mining Techniques* to minimize longer acronyms in the following chapters.

7. The process of identifying private attributes and correlation from a dataset that was previously anonymized.

8. *Data providers* are represented by IoT objects and belongs to *data owners*.

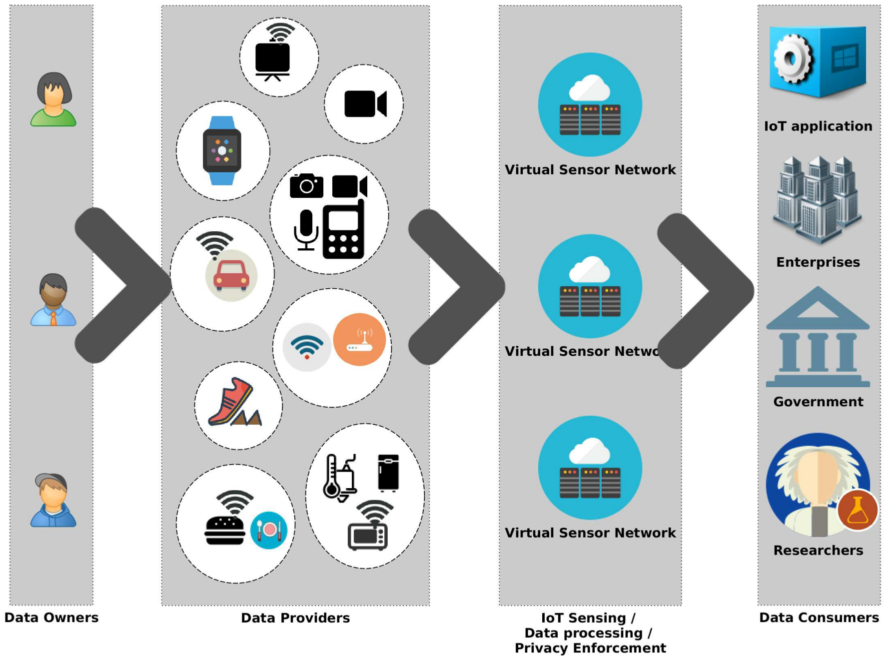

Figure 1.2 – Privacy-aware Sensing as a Service

to mitigate privacy risks related to privacy-sensitiveness of IoT sensing services.

— Defining an ontology for the Sensor Web that supports the classification of personal information based on behaviors (contexts), improving the representation of personal information to be classified and used to define privacy policies;

— A privacy-by-policy mechanism that prevents unintended and malicious inference by executing PETs selectively *on-the-fly* according to privacy policy conditions.

— Implementing a *privacy-aware Sensing as a Service* (*S^2aaS*) testbed in a real IoT platform based on our proposed privacy model and privacy-by-policy mechanism.

1.5.1 Privacy Model

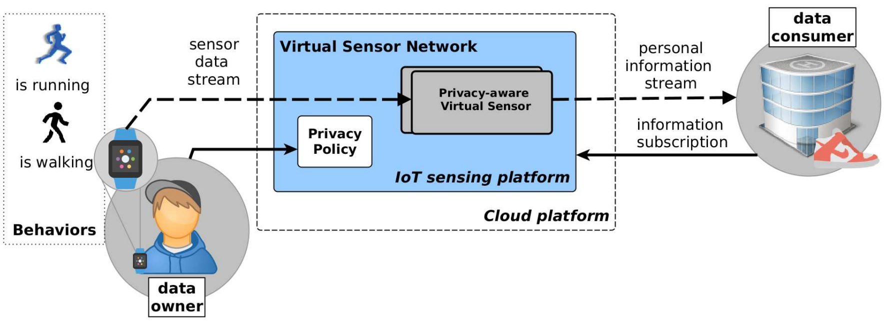

Figure 1.3 – Privacy-aware Sensing as a Service

The proposed approach for privacy preservation in the IoT is based on the IoT S^2aaS design that brings to an independent zone the data processing and the privacy enforcement. The goal of our proposed design is to shift the provision of privacy and data processing from the *recipient sphere* toward a *joint sphere* on which a privacy policy evaluation and enforcement are guaranteed between the *data owners* and *data consumers*. To this end, we designed the IoT S^2aaS based on VSN, Cloud Computing technology, and a *privacy-by-policy* mechanism. In this new model, the VSN delivers high-level information stream instead of sensor data stream, while enforcing privacy by evaluating *on-the-fly* preventively the inference intention and the produced personal information using the *privacy-by-policy* mechanism.

From the *data owner*'s perspective, her privacy policy is defined based on personal information and contexts, allowing the VSN to evaluate it for each sensor data stream from this *data owner*. From the *data consumer*'s perspective, the service model of the S^2aaS that provides information subscription will be restricted to high-level information. Additionally, *data consumers* (or outsourced independent developers) are able to specify, implement and execute their KDDM processes using the concept of *virtual sensors* that must be specified and deployed in the VSN to be executed and to provide the intended personal information.

The privacy model is based on the Sensor Web which employs semantic sensor annotations, such as those defined in the SSN-O, to provide services, such information retrieval, sensor interoperability, and sensor discovery using a high-level information about sensor, its characteristics, sensing conditions, platforms, observations and observed *feature of interest*. However, despite the fact that SSN-O is the most adopted and a World Wide Web Consortium (W3C) standard, its semantic representation for sensor and feature of interests are limited. To address this problem, we designed an ontology based on SSN-O, as described in the next section.

1.5.2 Ontology for Personal Information Classification on the Sensor Web

In the envisioned privacy model, *virtual sensors* and high-level *personal information* are key concepts. SSN-O represents personal information as *features of interest*, along with its observed properties, and sensors as anything that senses. Each feature has *properties* that can be observed by sensors which, in turn, produces observations. The domain-agnosticism of the SSN-O restricts the classification of a *feature of interest* as personal information due to its inability to represent its association with the *data owner*. Another limitation is the lack of representation for *virtual sensors* which are basically in-network data processing units, instead of sensors who detect stimuli to produce observations.

As part of our approach, we propose Ontology for Personal Information on the Sensor Web (OPIS), an Ontology for Personal Information on the Sensor Web, to represent personal information and *virtual sensors*. The proposed ontology aims to address shortcomings of

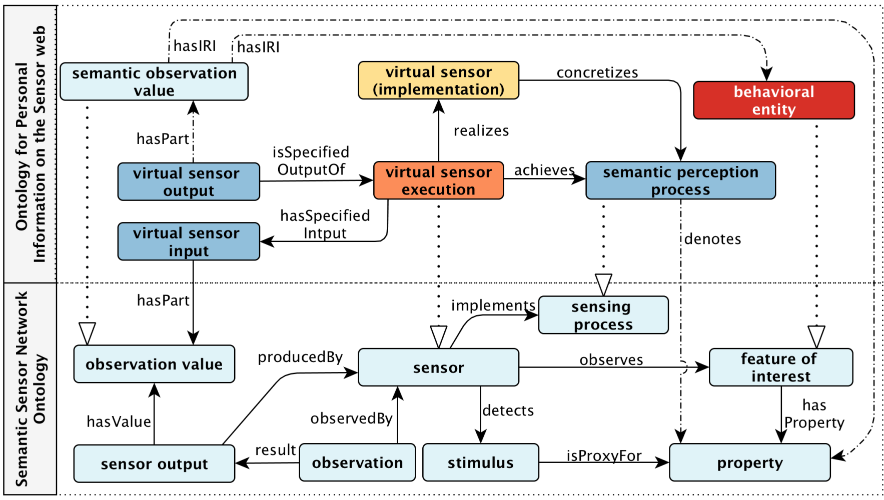

Figure 1.4 – OPIS overview

SSN-O regarding the representation of *virtual sensors* – KDDM process, implementation, and execution – and personal information – *features of interest* and observed properties. Our goal is to provide a modular ontology that extends the expressiveness of SSN-O and support the definition of privacy policies and the preventive evaluation of unintended inference as proposed in our privacy model.

To this end, we propose to extend the SSN-O concept of *feature of interest* to represent personal information, grounded on key concepts of the Behavior Computing (BC) [21], and the SSN-O concept of sensor into *virtual sensor*, providing semantic representation for KDDM process specification, implementation, and execution based on the Ontology of Data Mining (OntoDM). By extending the SSN-O, the proposed ontology is compatibility to IoT platforms and their sensor interoperability services.

1.5.3 Privay-aware Virtual Sensor Model: A Privacy-by-Policy Mechanism

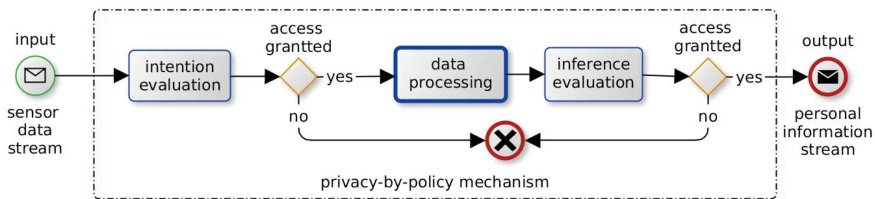

Figure 1.5 – Privacy-aware Sensing as a Service

Along with the privacy model, we developed a The proposed *privacy-by-policy* mechanism encapsulates the data processing with a two-fold Privacy Enforcement Point (PEP) that evaluates privacy policies with regard to inference intention and inference (data processing output), as depicted in Figure 1.5. The mechanism includes the data processing as part of the privacy enforcement, controlling the access of the input data flow or its data utility; as well as, the release of data processing output. Its goal is to improve the traditional mechanism by adding a preventive privacy policies evaluation based on the specified information to be produced by the data processing. This evaluation relies on the capability of the PEP to interpret the information that flows in and out of the data processing execution, and the KDDM process specification implemented in this data processing. We envision the preventive and selective execution of PETs, such as anonymization techniques or access denial, according to personal information classification; and the detection of malicious inference [9] by comparing KDDM processes.

To this end, we propose an ontology-based privacy mechanism that employs Semantic Web technology to represent and evaluate privacy policies and related information. In particular, these policies are defined by a set of Privacy Policy Conditions (PPCs), each of it composed by a personal information classification (antecedent) and a PET that should be executed (consequent). We define formally an extensible ontological framework to express these PPCs extending and reusing concepts of the Semantic Sensor Network Ontology (SSN-O) [10], OntoDM [11], and a domain specific ontology for personal information. This ontological framework benefits from the Ontology Web Language (OWL) reasoning capability to provide a flexible and powerful classification for personal information. The PEP is mainly composed by SPARQL queries to retrieve PETs and to compare similar KDDM specifications to detect malicious inference intention. Grounded on an analysis of privacy issues in the IoT, we propose to implement this model using the concept of *virtual sensor*.

1.5.4 Privacy-aware Sensing as a Service Testbed

Aiming to evaluate the viability of implementation for our approach, we provide an architecture for the privacy-aware S^2aaS, extending the eXtended Global Sensor Network (xGSN) platform [27], which configures an implementable testbed to verify the viability to implement the privacy model and the *privacy-by-policy* mechanism. Since our approach relies mainly on the reasoning capacity using OPIS, SSN-O, and the proposed ontological framework to define privacy policies, we implement SPARQL queries and instantiate classes of personal information, virtual sensors, and privacy policy conditions to evaluate the viability and response time of our proposed policy enforcement.

9. An inference is considered malicious when does not match its specification.
10. SSN-O is considered a *de facto* standards for semantic sensor annotation.
11. OntoDM is a meta mining ontology used to specify KDDM processes.

1.6 THESIS STRUCTURE

The remainder of this manuscript is organized in two parts: General Introduction and Contribution. Chapter 2 begins with a description of key concepts and enabling technologies of the IoT and the Cloud Computing that are important to understanding the privacy design choice of our approach. In order to define the guidelines and principles for the analysis of privacy approaches, we present key concepts of privacy engineering. We also present the most relevant works that address privacy in the IoT. Aiming to analyze privacy from different viewpoints, we discuss of IoT enabling technologies that impact in the strategy for privacy enforcement in the IoT. We elicited three perspectives – *sensor-centric*, *data-centric*, *human-centric* – that reflect the three main issues related to personal information in the IoT sensing.

In Chapter 3, we review the Semantic Web technology, ontologies for the Sensor Web, Meta-Mining (MM) and the theory of Behavior Computing (BC). We investigate shortcomings of the Semantic Sensor Network Ontology (SSN-O) with regard to its expressiveness to represent personal information and data processing techniques. In addition, we identify the Ontology of Data Mining (OntoDM) as the most suitable ontology to address the limitation of SSN-O to represent *virtual sensors* and KDDM processes. Next, we present key concepts of the BC and how it can be applied to address the problem of personal information representation in the behavioral context commonly found in situations of IoT sensing. Lastly, some related works for representation of personal information using ontologies are discussed, followed by conclusions of how these approaches can be employed to propose an ontology for personal information.

In Chapter 4, we review the two main Privacy-Enhancing Technologies (PETs): Privacy-Preserving Data Mining Techniques (PPDMTs) and Access Control Models (ACMs). These technologies provide privacy using different strategies. We investigate these approaches in order to understand how they address privacy preservation, analyzing them from a privacy engineering perspective and having in mind the privacy model that we intend to propose. Lastly, we conclude with a discussion about advantages and limitations of these works and how PPDMTs and ACMs could be incorporated to our proposed solution.

In Chapter 5, we present OPIS in details, setting formally class and property definitions. The ontology design process is also described, along with external imports and the realist vs descriptive views. The proposed ontology provides concepts to represent personal information in the IoT and virtual sensors, being the foundation to define privacy policies and Privacy-Enhancing Technologies (PETs).

In Chapter 6, we present the Privacy-aware Virtual Sensor Model (PA-VSM), our proposed *privacy-by-policy* mechanism implemented according to the *privacy model* based on the Cloud-IoT architecture. In this chapter, the ontological framework for Privacy Policy Condition (PPC) based on OPIS is formally defined, enabling an effective and extensible classification structure, a selective privacy preservation, and a user-friendly privacy policy definition. This ontological framework is used to define SPARQL queries and algorithms that form the foundation of our two-fold Privacy Enforcement Point (PEP), enforcing privacy preventively and belatedly.

In Chapter 7, we present a testbed for our approach by defining the implementable architecture of an IoT sensing platform – xGSN – extended to incorporate the PA-VSM. We demonstrate the usability and viability of OPIS by instantiating concepts to represent personal information and virtual sensors. In addition, the ontological framework proposed to be used as knowledge representation for the PA-VSM is also instantiated with PPCs. At last, we present the preliminary results of this experiment, the SPARQL queries used to query the Virtuoso triple store and their response times.

Finally, we summarize our contributions in Chapter 8, pointing the main contributions, drawbacks and future endeavors related to our approach.

Part I

GENERAL INTRODUCTION

2 INTERNET OF THINGS AND THE SENSING SERVICE

CONTENTS

INTRODUCTION

In this thesis, we propose an approach for privacy preservation in the Sensing as a Service (S^2aaS). In order to understand our motivations and research endeavor and to propose an implementable privacy model, it is necessary to understand basic concepts of the IoT, and its enabling technology and architecture. We introduce these concepts in Section 2.1. Due to the convergence of the IoT and the Cloud Computing service model, we include a brief explanation about the IoT architectures, middleware and its convergence with the Cloud-IoT paradigm and its sensing as a service model. In Section 2.2, we present privacy engineering principles, main concerns about privacy and preservation paradigms that will support the analysis of privacy-aware approaches and the design of our privacy model. Then, in Section 2.3, we describe IoT technologies from three viewpoints – device-centric, data-centric and human-centric perspectives – to facilitate the investigation of privacy issues in the complex Cloud-IoT infrastructure. In Section 2.4 we discuss a representative selection of approaches that deal with sensor data streaming, stream mining, stream reasoning, privacy preservation and access control for data streams in the IoT sensing service that support the definition of our *privacy by design* model. Lastly, in Section 2.5, we conclude by briefly summarizing the concepts of this chapter.

2.1 INTERNET OF THINGS

The term Internet of Things was initially coined to refer uniquely to identifiable interoperable interconnected objects equipped with Radio-Frequency IDentification (RFID) technology [28]. Today, along with the advances of IoT enabling technologies, the concept of IoT has evolved to a "dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual *things* have identities, physical attributes, and virtual personalities; using intelligent interfaces to be seamlessly integrated into the information network" [29].

The IoT has participated in the dissemination of pervasive computing through these identifiable, ubiquitous, autonomous objects, hereby called *IoT objects*, that effectively connect the real world to the virtual world. Thus, the IoT envisages a connected world where physical objects, beings, and information interact with each other seamlessly regardless of place and time. In [30], *IoT objects* are classified, according to design and architectural principles, in three dimensions: *awareness*, *representation*, and *interaction*. The *awareness* is the ability to understand (i.e., sense, interpret, and react to) real-world stimuli, including *human behaviors*. The *representation* refers to computational models on which *IoT objects* can be specified, such as programming languages or rules. The *interaction* denotes the ability to interact directly with other *IoT objects*, things or end-users.

The capacity to address uniquely a device or an identification tag through the Internet allows tracking this device or item wherever it is researchable through a network or communication channel, such as the Near Field Communication (NFC). More generally, the IoT objective of connecting *things* around the world supports the implementation of the ubiquitous computing and context-awareness [31].

2.1.1 IoT Enabling Technologies

The IoT architectural design is influenced by the assembled technology that enables IoT capabilities. Evidently, these technologies are not new, but together, they form a complex and interconnected infrastructure on which the IoT is based. On the other hand, the impulse caused by the IoT adoption also pushes the development of these enabling technologies. For example, the growing presence and scalability of wireless network access, such as Wireless Local Area Network (WLAN) and 4G-LTE, increases the IoT ubiquitousness. As a consequence, the development of the 5G is influenced by the IoT challenges and trends [32]. Figure 2.1 illustrates the main enabling technology incorporated in IoT which are discussed next.

IDENTIFICATION As previously mentioned, the initial concept of IoT was introduced when the RFID technology was presented, along with its identification and communication capacities. The RFID tag was

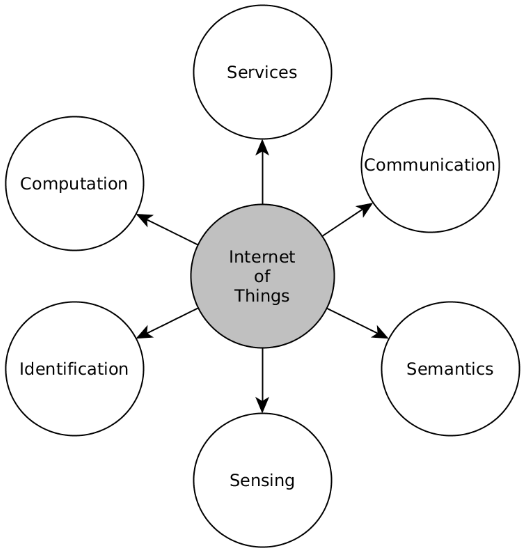

Figure 2.1 – IoT enabling technologies

crucial to enable IoT object discovery and unique identification, complementing the public and private networks limitation of global addressing [1]. From that on, several identification methods have been proposed, such as the open Standard Ubiquitous ID (SUID) [33], for uniquely identifying a data source or an IoT object. Moreover, identifiable objects are not limited to electronic devices and gadgets, but also includes non-electronic items, such as food, papers, official documents, works of art, equipment, and furniture.

SENSING The sensing technology has been claimed to be the major enabler of the IoT [34]. Its capacity to digitalize physical stimuli and to produce data that can be used to measure, aggregate, and perceive, provide insights about any observed feature of interest, changing the way systems interact with the things, situations and humans. The recent advances in sensing technology have achieved efficient low-cost sensors in large-scale production. As a consequence, this technology was widely adopted in personal devices, such as embedded sensors in smartphones (e.g, Global Positioning System (GPS), gyroscope, accelerometer, infrared) and in industrial environment sensors, such as a thermostat, smoking detector, and infrared surveillance cameras.

COMMUNICATION The variety of available communication technologies enables data exchange, connecting a wide range of heterogeneous *IoT objects*. Proprietary and open communication protocols have been proposed to operate in different conditions, such as power, noise, latency, distance, and so forth. In addition, communication liability is crucial to the IoT and is related to the network capability of self-adaptation and multi-path routing. Currently, wireless com-

1. The number of address of Internet address protocol Internet Protocol version 4 (IPv4) was limited and could not support the number of connected devices in the Internet. Its new version Internet Protocol version 6 (IPv6) fixes IPv4 addressing limitation.

munication technology is preferred to provide seamless integration of *IoT objects* in environments and mobile devices. Main communication protocols adopted in the IoT are: RFID, NFC, IEEE 802.11 (WLAN), IEEE 802.15.4 (ZigBee), IEEE 802.15.1 (Bluetooth), Multihop Wireless Sensor/Mesh Networks, IETF Low power Wireless Personal Area Network (6LoWPAN), Machine to Machine (M2M), 3G, 4G-LTE, IPv4, IPv6 [35, 36, 37].

COMPUTATION In order to provide smart services, the IoT has a computational capacity that is concretely implemented in its processing units, such as micro-controllers, microprocessors, and software applications [38]. Several IoT-friendly hardware technologies were conceptualized to suit the IoT, such as Raspberry PI [2], Arduino [3], and Galileo [4]. On top of these hardware technologies, modern Operating Systems (OSs) provides a richer real-time platform to control *IoT objects*, adapted to their resource constraints. For instance, Android and iOS are operating systems that provide a rich interface experience for the end-user, at the same time, that deliver computational capacity for mobile applications in smart-phones and smart gadgets. Additionally, smaller devices designed to work in limited computation capacity and low energy consuming require lighter weight OSs that efficiently balance functionalities and resource constraints, such as TinyOS [5], LiteOs [6] or RiotOS [7].

Despite their heterogeneity, the probable solution for privacy enforcement in the IoT should consider the memory, storage, and computational power limitations which part of the *IoT objects* have. This restriction has been addressed in the IoT by shifting resource demanding processing toward the network layer, where computational resources are more likely to be available. Allied to that, one of the main characteristics proposed by the IoT relies on the capacity to retrieve data from several *IoT objects* in order to respond coordinately to multiple sensing observations. Consequently, data aggregation and processing must take place somewhere other than the sensing layer. In this context, the virtualized and unlimited processing and storage capacity provided by Cloud Computing have been referred to deliver an effective solution for the IoT. In fact, IoT and Cloud Computing are argued to be complementary technologies [34]. Cloud Computing and its relationship to the IoT is described in more details in Section 2.1.3.

SERVICE Service refers to the implementation and management model oriented towards the quality of services and requirement specifications. The IoT paradigm is constructed on this concept of service where information retrieval and communication are accessed through services. Most of IoT architectures have been proposed based

2. https://www.raspberrypi.org/ (accessed on 26/04/2017)
3. https://www.arduino.cc/ (accessed on 26/04/2017)
4. https://software.intel.com/en-us/articles/when-to-use-the-intel-galileo-board (accessed on 26/04/2017)
5. http://www.tinyos.net/ (accessed on 26/04/2017)
6. http://www.liteos.net/ (accessed on 26/04/2017)
7. https://riot-os.org/ (accessed on 26/04/2017)

on the Service-Oriented Architecture (SOA) [31] which implements services by encapsulating the underlying complexity, heterogeneity, and technical details.

In [39], IoT services are categorized into four classes:

— *Identity-related*: services that consist in retrieving and communicating passively or actively to *IoT objects* based on its identity;
— *Information aggregation*: refers to services that aggregate and infer higher-level information based on sensor data collected from *IoT objects*;
— *Collaborative-aware*: services that use the information processed in information aggregation service to provide decision making information for end-users or to concretely ignite *IoT objects* to act (or react) according to a perceived situation.
— *Ubiquitous*: refers to the highest level of information reporting and IoT coordination, which provides autonomous, collaborative, pervasive computing any place, any time. The concept of *smart cities*, for example, is based on the ubiquitous services.

SEMANTICS The Semantic Web is a W3C effort to produce a formal knowledge representation to the World Wide Web (WWW) capable of expressing semantics for the information on the web. The W3C aims at standardizing concepts and encodings for the knowledge representation, shifting Web data towards a linked and semantic world, so called the Semantic Web. Semantics have been extensively used to improve the Machine to Machine (M2M) and Human to Machine (H2M) communications in IoT. In general, Semantic Web technology leverages IoT services enabling knowledge representation which is the key component to automate tasks in all IoT architecture layers and minimize the need for human intervention. Information discovery, sensor interoperability, data aggregation, information retrieval and complex event processing are a few examples of semantics potential to create more intelligent mechanisms that interpret conditions, contexts, data quality and sensor heterogeneity. Moreover, ontologies, such as the SSN-O [40], are used in the IoT to enrich the communication between *things*, and between *things* and IoT users.

2.1.2 IoT Architectures and Middlewares

Architectures aim to represent, modularize, and structure system functionalities. Many aspects and functions of IoT enabling technologies are combined to deliver the IoT paradigm. In fact, besides that, other issues related to the IoT exponential growth, security, accountability, and privacy have also been addressed in IoT architectural design [41, 42, 31].

Table 2.1 presents a mapping between abstract layers of proposed IoT architectures, as studied in [38]. The most basic model is conceptualized in a three-layer architecture and it was originally inspired by the network stacks. Although this classification does not comprise the current state of the art and complexity of the IoT, it provides us

Three-layer	Middleware based	SOA based	Five-layer
Application Layer	Application Layer	Applications	Business Layer
Network Layer	Middleware Layer	Service Composition	Application Layer
	Coordination Layer	Service Management	Service Management
	Backbone Network Layer	Object Abstraction	Communication Layer
Perception Layer	Existing Legacy Systems	Objects	Objects
	Access Layer		
	Edge Technology		

Table 2.1 – Mapping between IoT architecture abstraction layers

a starting point to understand the evolution of IoT architectures and how the network layer was specialized to handle its architectural dimensions.

In the lower level, the perception layer and its likewise pairs represent the pervasive part of the IoT that performs the real-world sensing and acting, through smart sensors and actuators; the access to data sources (e. g. existing legacy systems) and to edge technologies, such as routers and RFID reader.

In the middle level, the network and its respective specialized layers represent the intermediary process that receives data from the lower-level layers in order to make it available for storage, processing, and administration. It is worth remarking that the lower and mid-level layers communication protocols are designed to be complementary. Sensor networks tend to be composed of a high number of sensors. Therefore, protocols, gateways and wireless networks are typically represented in the backbone network and object abstraction layers. In order to abstract from this network protocol heterogeneity, the middleware, coordination, and service management layers are proposed to deliver Application Programming Interfaces (APIs) and services to interact with the underlying layers and coordinate these services.

In the higher level, the application and business layers represent APIs and services request by external applications or IoT end-users. IoT applications, such as smart cities and smart environments, are based on top of these layers that access the underlying middle-level services to retrieve aggregated information and interact with *IoT objects*. Since the application layer can be used to intermediate end-users and underlying layers through the business layer, in the five-layer architecture, the application layer is sometimes classified as intermediary or highest/end-user layer.

It is evident the influence that the service technologies have on IoT architectures. In fact, the key concept of service is based on the provisioning of common functions and services through published and discovered interfaces that typically encapsulates complexities and underlying heterogeneity. The service technology has been successfully employed in the software industry and, recently, have been used as building blocks of complex architectures and middlewares.

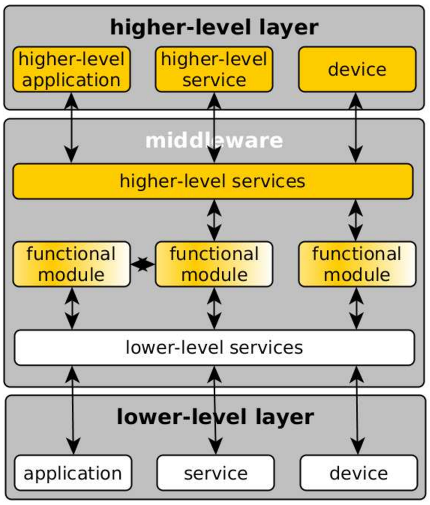

Figure 2.2 – Middleware structure

Middleware is a software layer that intermediate different level of complexities, as presented in Figure 2.2. Applications, services, and devices in different abstraction levels are intermediated by middlewares that provide interfaces for the higher and lower-level layers and a set of programming abstractions to implement the service provisioning. In the IoT, middlewares are commonly employed to facilitate the integration and the communication of heterogeneous data sources (e. g. *IoT objects*, edges technologies, and systems) [43]. The advantage of middlewares as an architectural design is the ability to incorporate, in a decoupled fashion, several issues and aspects that must be considered, such as interoperability, trust, scalability, security, privacy, extensibility, and so forth. Recent works in the IoT have confirmed the establishment of middleware as part of the IoT paradigm, comprising SOA, publish/subscribe mechanisms, semantic web technology, and Cloud Computing [44].

2.1.3 The Cloud Computing Service Model

The National Institute of Standard and Technologies (NIST)[8] defines Cloud Computing as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e. g. networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" [45]. The Cloud Computing has profoundly impacted the Information Technology (IT) sector and the way Internet services are delivered, targeting client's needs through a scalable and reliable information infrastructure. It was characterized by its successful *pay-as-you-go* model that

8. http://www.nist.gov/ (accessed on 26/04/2017)

allows clients to pay only the consumed resources while permitting service providers to manage resource costs and allocation.

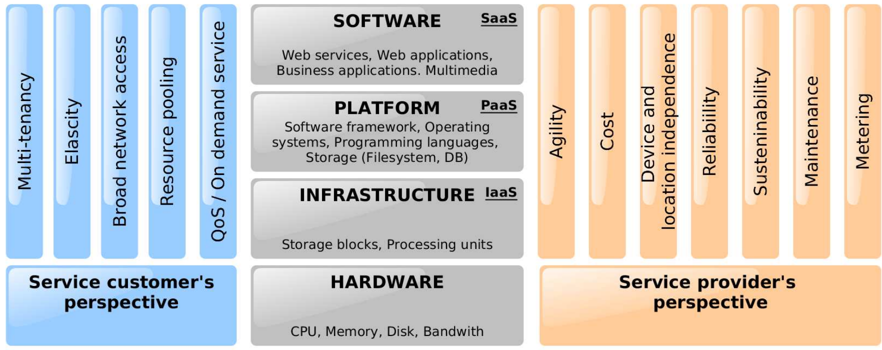

Figure 2.3 – Traditional cloud computing service layers and key characteristics

The advances in virtualization and *hypervisor* technologies for the Cloud service model enabled the dynamic resource allocation, providing elasticity (*on-the-fly* resource provisioning) and allowing the measurement and pricing of computational resource usage as commodities. Key characteristics of the Cloud Computing demonstrate how different is the Cloud model from traditional computing approaches, such as multi-tenancy, elasticity, on-demand service, cost, reliability, and so forth [46]. These characteristics are illustrated in Figure 2.3 divided into service customer's and provider's perspectives. Furthermore, services in the Cloud are traditionally classified in three layers [47]: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).

IaaS comprises computational infrastructures as data centers that provide storage and computing services. PaaS refers to an enterprise-grade cloud computing infrastructure that aims to provide the virtualized infrastructures for multiple tenants based on the physical infrastructure of computing nodes, storage units, and network that deliver services according to a Quality of Service (QoS) – performance, security, isolation. SaaS consists of software features that are made available on virtualized infrastructures according to a QoS. It represents the software functionality that is actually accessed, therefore, being the most visible part of the Cloud Computing for end-users.

The Cloud service model, so called Everything as a Service (XaaS), can be applied to any kind of service and technology layers [2]. In particular, the provisioning of IoT service through the Cloud infrastructure and service model have been evident in the increasing number of new services paradigms created by the integration of the Cloud and the IoT, such as Thing as a Service (TaaS), Database as a Service (DBaaS), Data as a Service (DaaS), Identity and Policy Management as a Service (IPMaaS), Video Surveillance as a Service (VSaaS), S^2aaS , Sensing and Actuation as a Service (SAaaS), Sensor as a Service (SEaaS), and so forth [48, 34].

The Cloud Computing has been employed in the IoT as an intermediate layer between *IoT objects* and IoT users, shifting the com-

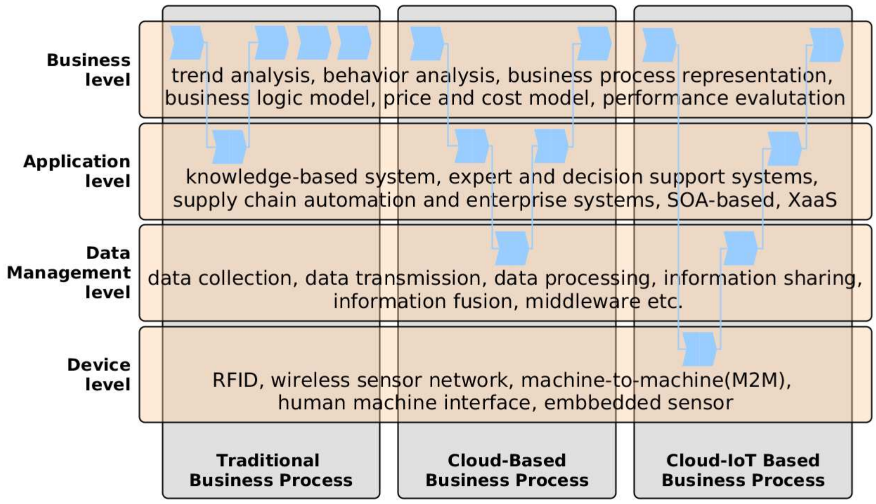

Figure 2.4 – Comparison of the business process perspective in the traditional, cloud-based and sensor-cloud-based scenarios.

plexity, maintenance and risk of information infrastructures towards the infrastructure provider – generally the IT sector – who is better equipped to manage it [34]. Indeed, the Cloud facilitates the flow of the IoT data collection and provides data processing capacity, while billing the cloud customer (IoT user) according to the amount of data collected, transmitted, processed and stored. Furthermore, many technological issues faced by the IoT have been partly addressed by the Cloud Computing, such as scalability, interoperability, reliability, efficiency, availability, and security [36]. On the other hand, the Cloud benefits from IoT real-world and pervasive scenarios that promote its adoption [34].

This novel paradigm called Cloud-IoT emerges from a synergy of complementary technological aspects, such as displacement (pervasive *vs.* centralized), reachability (limited *vs.* ubiquitous), components (real world things *vs.* virtual resources), computational capabilities (limited *vs.* unlimited), big data (source *vs.* means to manage) [34].

This disruptive model has a profound impact on business and personal computing. While businesses are highly interested in the information chain that can be ignited through the Cloud-IoT, individuals are concerned about information ownership, participation and how IoT services are seamlessly delivered to them. Business applications can carry on trend analysis, behavior observation, benefit identification, business process representation, business logic modeling, price and cost modeling, performance evaluation; ultimately igniting a faster information chain by incorporating the IoT sensing service [49]. It is evident the added value and real-timing that the Sensor-Cloud delivers to the business and its business process information chain. Figure 2.4 illustrates a comparison between different business process scenarios and the type of service delivered at each IoT architecture layer. From traditional to Cloud-IoT based business processes, the technological incorporation level stepped from a simple application usage to a complex interconnected process that communicates and integrates directly to anything (real world devices and virtual resources), relying on data management capabilities provided by the Cloud infrastructure.

2.2 PRIVACY ENGINEERING

The role of personal data protection principles is to support the maintenance of trust in the continued usage and benefits of personal information. After In information systems, these principles emerged in the form of guidelines, such as privacy-by-design, privacy impact assessment, and data breach notification, due to the convergence of best practices and the commitment of different actors, stakeholders, and advocates on privacy protection [50].

The concept of privacy-by-design, in particular, includes a holistic and robust approach to address the systematic effects of Information and Communications Technology (ICT) and large-scale networked infrastructure, which has been particularly witnessed with the expansion of the IoT. It started to gain attention in 2010 [51], at the International Conference of Data Protection and Privacy Commissioners, recognizing that privacy should be inherently embedded into architecture design, operations, management of ICT systems along with the entire information life cycle. Its foundational principles include proactivity, privacy as the default, privacy embedded into the design, full functionality, end-to-end life cycle protection, transparency, and respect for user privacy.

Practically, privacy-by-design is a holistic paradigm that considers preventively Privacy-Enhancing Technologies (PETs), processes, and practices into the system architecture in order to protect privacy seamlessly [50]. Besides that, it includes guidelines that emphasize the importance of thinking about privacy protection through the architecture design. Langheinrich [52] emphasizes six main areas of system design that should consider in ubiquitous computing and ICT system development: i) notice of data usage, collection, and inference (transparency); ii) choice and consent; iii) de-identification [9], which guarantees data utility and usability not linkable to the concerned individuals; iv) proximity and locality that constitute a common criteria between physical and information privacy; v) adequate security; and vi) data minimization that restricts data collection and usage for a well-defined purpose (no "in advance" storage or unnecessary data collection).

According to privacy-by-design guidelines, PETs should be included in the IoT system as part of their architectures and privacy preservation strategies since its conception. It is worth to mention that these guidelines should be considered during the process definition in different levels, such as business process, system workflow,

9. In this manuscript, the terms "de-identification", "pseudomyzation", and "anonymization" are used interchangeably. The focus is on the classification of these techniques instead of the importance in the subtle differences between them.

and PET procedures, algorithms and development patterns. Different types of PETs can be combined or used separately to preserve privacy in several contexts, such as those of security multi-party computation, homomorphic encryption, private information retrieval, anonymous credentials, and communication, trusted environments and platforms [53]. Not surprisingly, security is inherent to privacy considerations because of the need for a safe environment and technology upon which privacy protection and trusted communication can be ensured. However, in this manuscript, although we identify security as an indispensable component in privacy-aware systems, which deserves attention and further investigation, we solely focus on the privacy protection of IoT systems.

Figure 2.5 depicts the orthogonality among privacy-by-design guidelines and components of privacy-aware system architecture; illustrating the plurality of PETs that may be adopted by one single systems and how the privacy-by-design guidelines can influence the development of different strategies for the same type of PET, such as security multi-part computation, right to be forgotten, access control based on attribute, role, or social network criteria.

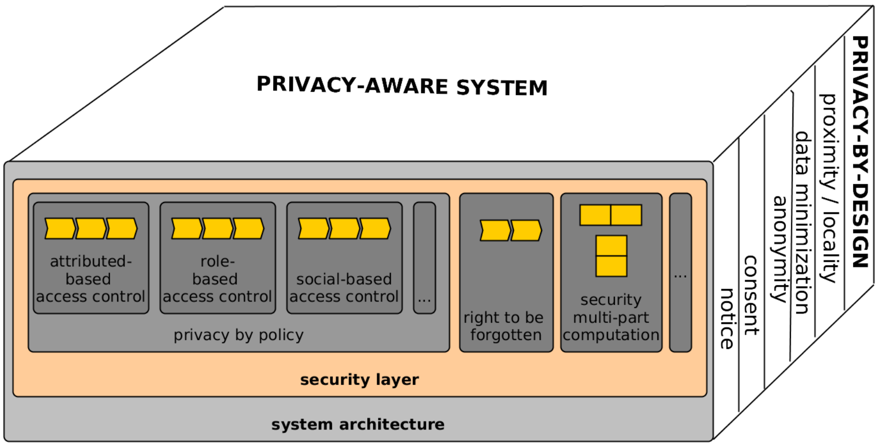

Figure 2.5 – Privacy-by-design strategy overview

The principles of openness on which the concept of *notice* is defined are the starting point of individual's awareness about her own privacy. The current privacy control provided by IoT objects and smartphones are not able to present the real privacy threat of disclosing specific information, such geographic location sensor data or list of contacts. The informed *consent* or agreement should be based on the full exposure of the facts the concerned individual needs to make a decision intelligently, including awareness of risks, usage, and alternatives to provide personal data [54]. Mechanisms based on privacy-by-policy, for example, should ensure compliance with requirements defined by privacy regulation, privacy risk management, and individual's choices on the consent of data subject, access and usage from an individual-centric perspective. In an opposite direction, IoT systems collect sensitive data and information about an individual, her identity and behaviors continuously, providing privacy control remotely from the individual's control zone. In practice, privacy-by-policy activities are often limited to the fulfillment of compliance requirements, which are pre-defined by the *data consumer*, providing little effective protection [22]. In other words, the fact that IoT systems are adopting different privacy-by-policy PETs do not guarantee that it is designed to protect the individual's privacy or that implements the designed privacy strategy efficiently. In addition, the intensive data flow of ICT and IoT systems – from portable devices, environment sensors, and personal information systems to third-party servers, owned by *data consumers*, – increases this doubt by collecting data indiscriminately, without notice nor consent.

The *de-identification* techniques aim at removing personal information, or its association, between a set of identifying data and the data subject, that can be the individual's identity or a sensitive data. The collect of techniques is vast and they vary in terms of levels of effectiveness and data utility. The great controversy around its efficacy has been argued due to the possibility of *re-identification*. For example, *privacy-by-architecture* that incorporate anonymization natively can suffer from chronic issues in these techniques related to data quality and background knowledge-based attacks.

Among these technologies, anonymization techniques are commonly employed to preserve privacy in the IoT sensing services, since it is built based on the premises that after anonymized, personal data could be released. In the remarkable work of Paul Ohm [25], he argues that anonymization has failed in the *anonymize-release-forget* paradigm. The process of *anonymization-re-identification* (or *anonymization-deanonymization*) has become a cyclic process, where anonymization methods intend to pointedly erase traces of sensitive information, respecting a level of *data utility*, while *re-identification* (or *deanonymization*) pushes the inference capacity of KDDM [10] algorithms further as a countermeasure. Thus, once a more powerful anonymization technique is proposed, it is commonly followed by the development of more effective KDDM workflow to *re-identify* sensitive information. However, while KDDM workflows have the possibility to aggregate unlimited background knowledge gathered elsewhere to increase its rate of *re-identification*, anonymization is limited to modify the original data set having in mind its data utility. In other words, the anonymized data set can become useless if it is transformed in a way to hide all possible sensitive information.

On the other hand, anonymization suits eventually the trade-off between system usability of IoT sensing services and privacy concerns, since most of these techniques offer a mechanism to control the level of data utility to be guaranteed in detriment of privacy preservation.

Gürses and del Alamo [22] claim that purely technical approaches to privacy preservation *might prove insufficient for aligning nuanced legal policies with engineering artifacts*, pointing to three principles of privacy engineering that must be observed:

10. KDDM concerns the entire knowledge extraction process which include data mining, machine learning and knowledge discovery research domain. Further information in Section 3.

— *Plurality*: refers the transversal classification for the diversity of approaches to privacy, such as those based on policy, social affinity, intimacy, trust, economic trade-off or simply *the right to be forgotten* paradigm.
— *Contextuality*: reflects the context-dependence of privacy preferences where the same person can be inattentive about certain information in some situations, and extremely concerned about privacy issues in others.
— *Contestability*: consists in the existence of privacy conditions, which can be evaluated and contest, if multiple interpretations or conflicting privacy requirements happen.

Therefore, the data utility of IoT sensing data or the PET could be adapted according to the risk of privacy harm. Some aspects of the use and processing of personal information can minimize the chances of privacy attacks. Ohm [25] highlights five characteristics that should be considered while designing a strategy to address these privacy harms:

— *data processing techniques*: privacy attacks should be anticipated and prevented based on data mining techniques. *Re-identification* and unintended inference intentions should be detected based on KDDM specifications, implementation, and workflows in order to adapt the privacy strategy;
— *private release*: public data releases are more likely exposed than private data exchange. Private data release offers better conditions to preserve privacy, and privacy accountability (data breach traceability);
— *data quantity*: privacy adversaries tend to find it easier to employ background knowledge by increasing the level of confidence of their inferences using a large amount of data. By controlling unnecessary data stream or data release, privacy harms can be minimized;
— *motive*: declared *re-identification* intention is commonly announced for private or governmental research purposes, supporting the user decision making to disclosure datasets or data streams. The declaration of usage intention for data release can be a good practice and may serve as evidence for lawful dispute related to privacy harms;
— *trust*: trust in *data consumers* and their reputations can server as parameters to assess privacy risk. In this context, social network and affinity degrees can be used to approximate their level of trustworthiness.

Privacy-by-policy mechanisms, on the other side, are designed to allow individuals to choose what to disclose, considering different types of criteria that can vary from data type, proximity, or social context. However, most of the time these mechanisms are provided by *data consumers* who, ultimately, have the major interest in extracting valuable personal information from collected data. In the end, the bad or unintentional design of system and technology, that cannot be legally accounted or assessed, can expose personal identity and information, defeating any intention of individual's privacy awareness or policy enforcement.

Given current IoT architectures, it is evident that traditional system engineering methods, in particular, conventional *de-identification* and *privacy-by-policy* PETs only covers partly these privacy engineering principles.

Part of the responsibility for privacy risks and breaches is under engineers, which may adopt these *privacy-by-design* guidelines and privacy engineering principles to minimize privacy intrusions. Nonetheless, the exercise of selective control of access to the self (consent) and the awareness of the potential consequences of this exercise (notice) rely on *data owners* and *data consumers*. These boundaries of (legal and accounting) responsibility, trust, and influence in ICT and IoT systems can be classified into three distinct spheres of influence [24]:

— *User sphere*: refers to the user's zone of control and responsibility to store, process and share data, relying on the individual's expertise to control access through the use of Privacy-friendly Information Systems (PISs) and PETs in personal computers, mobile devices, and, recently, computational resources provided by the Cloud.

— *Recipient sphere*: consists in the zone of the *data consumer*'s control where the offered back-end infrastructure (which can also be virtualized in the Cloud) provides services to collect, storage, and process data, minimizing privacy risks through security and privacy mechanisms.

— *Joint sphere*: denotes to a third-party zone of responsibility, trusted by the *data consumer* to intermediate the data collection, processing and sharing, and which offers an independent service that allows *data consumers* to protect their privacy while committing levels of quality of service to *data consumers*.

The *joint sphere* is compelling due to the outsourcing of data collection, processing, storage and privacy enforcement to a third independent party. By shifting the data operations from the *recipient sphere* to the *joint sphere*, independent trustee parties can take over these responsibilities without compromising individual's privacy, once information would not be its core business, but rather the provision of services to empower individuals to administrate their own information.

2.3 IOT PERSPECTIVES

The ethical issues associated with the social and personal impact of these information-centric technologies limits the confidence in and acceptance of the IoT by individuals who participate actively or passively in this information chain. In order to investigate the privacy issues in the IoT sensing service based on the privacy engineering principles and privacy harm risks, we propose to examine the IoT from three main perspectives: *device-centric* [55], *data-centric* [56, 57], and *human-centric* [58, 59].

Originally the IoT was built based on a *device-centric* perspective that focused on establishing the technology needed to integrate and manage devices. From this point, more complex bottlenecks related to the capacity of these devices for *data processing*, *data storage*, *sensor discovery* and *interoperability* support our conclusion to propose a privacy preservation strategy towards *device virtualization*. Besides that, the Cloud-IoT paradigm from a *data-centric* perspective aggregates approaches proposed to solve data streaming, aggregation, storage, search, and analytics issues. The *data stream*, Complex Event Processing (CEP) and Semantic Stream Reasoning (SSR) are important concepts that enable our proposed privacy preserving mechanism. Finally, the intrinsic human involvement in the current IoT sensing scenarios needed to be inspected, along with *information ownership* and *intention of data usage* notions. These concepts will permit us to understand why current approaches to personal information classification are insufficient to address current privacy threats in the IoT sensing service. Table 2.2 presents the sensor, data, and human-centric perspectives, its key concepts, characteristics and issues that are detailed in the following subsections.

2.3.1 Device-centric Perspective

In the device-centric perspective, the integration and management capabilities of sensing resources are crucial features that aim to minimize the device heterogeneity issue typically found in the IoT [55].

INTEGRATION AND MANAGEMENT CAPABILITIES Many *sensor network protocols and standards* are proposed to address sensor integration to the IoT network (see Section 2.1.1). These protocols and standards are part of the communication technology that aims to deliver reliable and safe communication channels to enable seamlessly user enrollment and device integration in the sensing service scenarios. Another key feature to integrate sensors in the IoT is its capacity to abstract *sensor interfaces* and *sensor data management* capabilities from heterogeneous hardware solutions, offering standard interfaces that intermediate the access, interaction and communication with sensors [55].

Recent review studies and surveys have indicated challenges related to the sensing service, such as availability, reliability, mobility, sensor density, sensor distribution, scalability [38, 76]. The degree of human intervention can also represent an excessive presence of *user sphere* approaches, which increases the human error propensity in the sensing, data processing human participation, and an exceeding user intervention for security and privacy enforcement. Low human intervention, therefore, should be compensated with *IoT object self-* capabilities*, among which the most notable are self-adaptation, self-organization, self-reaction, and self-processing [77].

Security can refer to data transfer, storage or processing. As we analyze the sensor integration capability, security protocols are intensively investigated in the perception layer and communication layer

Perspective	Key concept	Issues & Characteristics
device-centric	integration and managament capacilities [60, 61, 55, 38]	sensor network protocols and standards
		sensor interface, sensor data management
		availability, reliability, mobility, sensor density sensor distribution, scalability
		self-capabilities
		security and privacy
	system partitioning scheme [62, 63, 64, 26]	in-device
		in-network
		in-cloud virtualized
	sensor discovery and interoperability [60, 40, 65]	mobility and geographic localization
		quality of service
		sensor provenance
		security and privacy
data-centric	data taxonomy [66, 56]	data generation
		data interoperability
		data quality
	data streams [56]	general and identifiers (RFID) data stream processing
		semantic stream reasoning
	data analytics [67, 68, 69, 70, 71, 26]	data stream mining
		complex event processing(CEP)
	data storage [72, 26, 56]	large-scale storage in distributed environments
		storage on resource-constrained devices
	data search and aggregation [70]	deep web and semantic web
		web search (linked data)
human-centric	information ownership [73]	personal and household
		private organizations public organization
		commercial sensor data providers
	participatory sensing [74]	humans as targets of sensing
		humans as sensor operators
		human as data sources
	opportunistic sensing [59]	user awareness / human behavior
		ambient awareness / contextual environment
		social awareness / social context
	sensing scale [75]	individual sensing
		group sensing
		community sensing
	participation in data fusion and data analytics [60, 74]	human intelligence / crowd-sourcing
		semantic and cognitive perception
	data usage intention [75]	learn, inform, share, persuade, act research purposes

Table 2.2 – Perspectives and characteristics of the Cloud-IoT

of the IoT. Despite the fact that *security* is a basic feature to guarantee *privacy*, the former refers essentially to access authentication and authorization, while the latter is related to the identity and personal information protection, i. e. it is able to distinguish private content. At the perception layer, *security* and *privacy* must be implemented in hardware firmware with secure communication protocols that establish secure transmission channels and encryption.

SYSTEM PARTITIONING SCHEME Originally, the IoT network layer stratification was not developed in different sub-layers, which consequently presented two *system partitioning schemes*: *in-device* or *in-network*. System partitioning refers to the distribution of data processing tasks and storage in different computational layers. The heterogeneity of *IoT objects* in terms of resource limitations can be gracefully addressed by system partitioning systems that are able to balance their resource restrictions, their intensive processing tasks, and the IoT infrastructure [62].

Recently, in accordance with the trend of Virtual Sensor Network (VSN) [63] and its software-defined management [78] in the Cloud-IoT paradigm, system partitioning approaches towards *device virtualization* have gained the attention of the industry and research communities [64, 27, 26, 79, 80]. *Device virtualization* delivers a higher-level layer to provide sensor integration, interoperability, and management [55]. *In-cloud virtualized partitioning scheme* refers to approaches that address the IoT sensing *data-centric* issues (filtering, processing, and storing) from a *device-centric* perspective. In fact, as Cloud Computing evolves, so does the Cloud-IoT sensing service in terms of efficiency to deal with *data-centric* problems. Therefore, shifting towards *device virtualization* is strategically relevant to incorporate the advances in the Cloud Computing technology. Additionally, the Cloud-IoT paradigm also offers the advantage to implement *joint sphere* solutions for privacy, addressing technological, informational and human-centric issues by intermediating these perspectives with services.

SENSOR DISCOVERY AND SENSOR INTEROPERABILITY *Service discovery* is a fundamental service as other upper-level services highly rely on its result to provide service composition, provisioning, and *sensor interoperability* [81]. The device heterogeneity in the IoT makes necessary to define a minimum set of functions implemented by sensors in order to provide *sensor discovery* and *sensor interoperability*. Ontologies, such as the SSN-O [40] and SemSOS [82], are used in these approaches, allowing semantic representation of the sensor capacity and semantic enrichment for IoT sensor data. Several criteria can be defined semantically about devices to ignite these services, such as *mobility*, *quality of service*, *sensor provenance*, *security, and privacy*. As the number of geographic location embedded sensor increases, *mobility and geographic localization* are commonly used to retrieve sensor. The *quality of service*, such as network latency, bandwidth, availability, and device processing capabilities can be represented using semantics, allowing fine-grained criteria for *sensor discovery*. *Sensor provenance* refers to the representation of sensing features, data generation, data processing, and involved agents that can be used to assist data consumers to understand, verify, and assess the data quality, and its trustworthiness [83]. Therefore, *sensor provenance* can be used to implement *data processing*-based privacy mechanism to anticipate inference intention. However, current ontologies for *sensor discovery* and *sensor interoperability*, such as SSN-O, are not expressive enough to represent *device virtualization* and its KDDM-based processing capacities. In terms of *security and privacy*, *sensor discovery* approaches do not consider *trust* as part of its criteria and, therefore, are unable to compare trustworthy sensors.

2.3.2 Data-centric Perspective

The *data-centric* viewpoint focuses on issues related to data taxonomy and the consequence of the Cloud-IoT as a big data major driving force. From a *data-centric* perspective, *data taxonomy*, *data stream*, *data storage*, *data search*, *data aggregation*, and *data analytics* are critical for effective IoT sensing enablement. The *data taxonomy* can be described according to its generation, quality, and interoperability [56].

DATA TAXONOMY The amount of data generated in IoT have different rates (velocity), scale (volume), heterogeneity (variety), and dynamics (mobility and geographic location distribution) [66]. These characteristics of *data generation* are commonly aligned to big data challenges, which are typically addressed by the Cloud infrastructure. Another important aspect is the *data interoperability*. As explained in the previous section, the semantic enrichment plays an important role in this aspect, adding a complementary information to the collected raw data, and modeling its meaning. Lastly, the *data quality* aspects constitute an important characteristic of the *data taxonomy*, such as precision and probability distribution. Due to the heterogeneity of *IoT objects*, common *data quality* issues includes uncertainty, redundancy, ambiguity, and inconsistency.

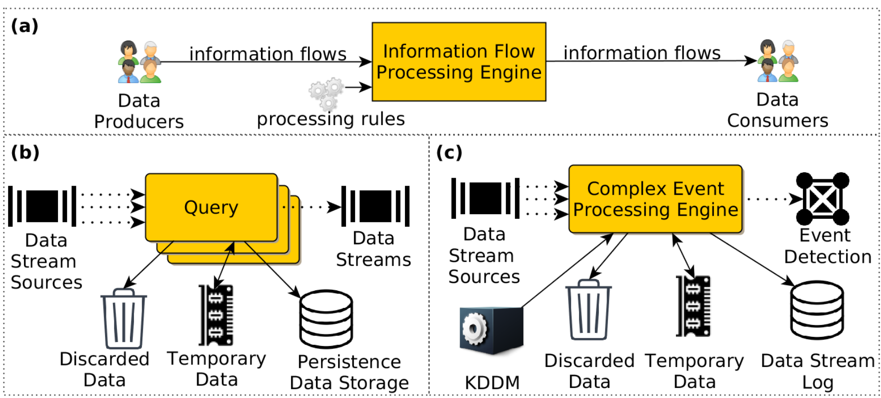

Figure 2.6 – Overview of information flow processes. (a) Generic model. (b) Data stream model. (c) Complex event processing model.

DATA STREAM The continuous data generation in the IoT sensing can be classified as a *data stream*. Figure 2.6 presents an overview comparison between (a) generic information flows, (b) *data stream* and (c) Complex Event Processing (CEP). A *data stream* is a sequence of timestamped data packages which are potentially unbounded; and characterized by continuous data arrival, data window size, and variable probability distribution [56]. Instead of asking for updated information, Data Stream Management Systems (DSMSs) provide active notification based on predefined queries. These queries are typically defined using a query language derived from SQL, including instructions to specify *windows* and to isolate portions of input streams to be converted into relational tables or to compose another data stream [84]. In Figure 2.6.(b), this scheme is depicted using several input streams, which are submitted to different queries, resulting in an output stream or stored data. The temporary memory illustrates the system working memory used to support aggregation operations, while discarded data represents unwanted data that has not been captured by query criteria. Each data package in the input stream can be an atomic value or a multi-dimensional attribute vector of atomic values. In general, raw sensor data and semantic annotations constitute the main streaming of data packages in the IoT.

Several challenges are related to *data streams*. Devices should consider its own limitation (battery, processing, and storage) and the IoT network layers, such as gateways and local connection bandwidth, to properly implement data streaming. RFID data streaming consists in a particular case of *data stream* for being one of the IoT cornerstones. Ambiguous, incomplete or missing identifiers may obfuscate the data streaming since it precludes IoT object traceability. Besides that, the data volume generated by the IoT object identification demands effective compressions techniques to minimize the network and storage consumption. Resource Description Framework (RDF) [11] triple stream processing issues arise as a consequence of the semantic enrichment of sensor data. Semantic Stream Reasoning approaches are proposed to query and reason about semantic annotation in the *data stream* in real-time, such as C-SPARQL [85], EP-SPARQL [86], CQELS [87].

DATA ANALYTICS The extraction and transformation of information from sensor data are implemented using Knowledge Discovery and Data Mining (KDDM) techniques. The continuous nature of *data stream* requires that data processing operations are executed *on-the-fly*. This process is known as Data Stream Mining (DSM) and intends to extract patterns and trends from data streams. Many data mining techniques have been implemented natively in queries mechanisms, supporting powerful data mining tasks, such as clustering, classification, and outliers detection [71, 67]. In this same direction, Complex Event Processing (CEP) consists in associating a precise semantics to the result of filtering, aggregating and mining the sensor data [88] (see Figure 2.6.(c)). KDDM techniques are extensively employed in CEP to generate

11. The RDF is general knowledge modeling framework designed by the W3C to specify metadata of resources in the WWW. Further information in Chapter 3.

abstractions that can be associated with background knowledge and, consequently, generate higher-level information based on abductive-deductive reasoning [12] [89]. More recently, particular CEP processes, such as Semantic Perception (SP) [68, 90, 58] and Perception Computing (PC) [69], have been proposed, aiming at delivering perception and learning capacity to the IoT.

DATA STORAGE The traditional database management systems are not adapted to the Cloud-IoT big data scenario. The *data storage* in the IoT has two opposite concerns: big data and device storage constraints. *Large-scale storages in distributed environment* approaches have been proposed to support big data scenarios. In fact, the Cloud infrastructure behind the IoT sensing service addresses this issues, providing optimized distributed storage systems that guarantee consistency, availability, and partition-tolerance. More recently, innovative approaches that take advantage of the IoT characteristics, such as mobility, mutual information interest, and unique identification, have been proposed to provide more efficient data processing and storage [72, 26].

DATA SEARCH AND DATA AGGREGATION The task of resource search in the IoT is a challenge as the Internet search implemented by the main searching engines (Google, Bing). Structured and semi-structure types of information in the WWW are indexed and semantically annotated, aiming to provide the best match for end-users' search request. The Deep Web targets at semi-structured information search, using Semantic Web technology (RDF and OWL) to enrich sensor data. By holding indexed and RDF-based information, web search mechanisms, such as real-time web search, RDF-based data search, and collaborative web search are proposed in the IoT to minimize the searching problem in a virtual space with billions of connected *things*. Initiatives to incorporate semantic and interlink public datasets use the Semantic Web to expose, share, and connect information from diverse areas are adopted in the IoT by some approaches [91, 92].

2.3.3 Human-centric Perspective

The *human-centric* perspective investigates the relations and roles that people have in the IoT sensing. As the density of IoT object per physical spaces increases, the IoT becomes more personal and social through the proliferation of mobile phones, smart gadgets, and social network based services. The human involvement in the Cloud-IoT is not limited to the role of information consumer. In fact, the distinguishing aspect of *human-centric* in the IoT sensing service is transversal in the data-to-decision or data-to-action paths.

12. Abductive inference or abductive reasoning seeks the most likely explaining from a set of observation. Deductive reasoning or deductive inference has a guaranteed conclusion based on premises and observations.

INFORMATION OWNERSHIP An IoT object is owned by an agent (individual or organization) at a given time. Hence, the information ownership produced by (or related to) these *IoT objects* may change over time. In [73], the IoT object ownership is classified in *personal and household*, *private organizations*, *public organizations*, and *commercial sensor data providers*. A personal item, such as mobile phone, smart watch, or tagged work of art belongs to the *personal and household* category. All *IoT objects*, including sensors, which are not owned by the public or private sector should be classified as personal. *Private organizations* and *public organizations* categories consist of buildings, facilities, and other traceable assets that are part of the IoT and owned by private companies or the government. The main difference between them is that public assets and infrastructures, such as bridges, roads, and parks are normally of public domain and, therefore, different rules must be applied in terms of security and privacy. The *commercial sensor data providers* are business entities that distribute, deploy, and manage *IoT objects* by keeping ownership. Mostly, they focus on public and private spaces where sensing is economically or strategically valued.

PARTICIPATORY SENSING In the perception process, the human involvement fills a mobility gap that the traditional sensing lacks, being able to follow the dynamic nature of crowds and covering areas where normally continuous fixed sensing are economically unfeasible. There are two paradigms of human involvement in the perception process [75]: *participatory sensing* and *opportunistic sensing*. In the *participatory sensing*, the agent who owns the IoT object actively engages in the data collection activity. It addresses information dissemination and sharing withing and among opportunistic communities (equipped with sensing devices) that are formed based on the movement and opportunistic contact nature of humans [59]. The *opportunistic sensing* does not depend on agents' involvement and automatically collect information based on its awareness.

The *participatory sensing* can be classified in [74]: *humans as targets of sensing*, *humans as sensor operators*, and *humans as data sources*. The most obvious and explored category is *humans as targets of sensing* that uses embedded sensors and applications to collect data to infer human activities, crowd behaviors, behavior patterns, etc. The *humans as sensor operators* search topic investigates how people can become part of the sensing infrastructure by allowing access to their mobile applications and embedded sensors and, consequently, expand the sensing network. The *humans as data sources* category refers to human knowledge and human intelligence as sensing. For example, the human participation in the recognition and identification of people and faces in a picture constitutes a type of human sensing.

OPPORTUNISTIC SENSING In the *opportunistic sensing*, the data collection is automated with no user involvement. In fact, the context is perceived in order to activate opportunistically the sensing based on *user awareness*, *ambient awareness*, and *social awareness*. *User awareness* concerns the ability to understand individual's context and behavior. *Ambient awareness* refers to the environment perception and situations. *Social awareness* consist of social contexts on which an individual is, such grouped activities, social events, and friend detection.

SENSING SCALE Allied to the capacity to perceive social interactions, the sensing scale can target specific *individuals sensing*, *group sensing*, or *community sensing* [75]. The *individual sensing* consists of data collection and analysis of a single specific individual. *Group sensing* is related to a social network of individuals who shares common interests and characteristics, such as neighborhood safety or body sensing of a soccer team. *Community sensing* refers to a large number of individuals who participates in the sensing normally used to infer crowd patterns, such as the spread of disease across a geographic region and daily transport urban migration. The *opportunistic sensing* can facilitate the adoption of participation in IoT sensing by automatic adaptation t the user context. Without user intervention, sensing services can efficiently use computational resources only in situations when data collection is needed, considering restrictions imposed by users. However, the lack of privacy preserving mechanism to intermediate IoT services is still insufficient to create data producers' confidence and foment user participation [4].

PARTICIPATION IN DATA FUSION AND DATA ANALYTICS The human participation in *data fusion* and *data analytics* consists in incorporating the *human intelligence* in the data collection, data analysis, and result visualization, leveraging the IoT capacity in tasks that are normally better executed by humans than by computers. *Crowdsourcing* platforms, such as Amazon Turk [13] and CrowdSource [14], implements this human-machine processing paradigm. However, this paradigm (and available tools) usually privileges the data consumer, offering data processing and data management collecting further information about the involved individual to increase the level of confidence and liability in produced information. In the Cloud-IoT, this type of human participation in data processing tasks can be realized using the application layer or the direct interaction with smart objects in the real world.

The perception and cognition of the world through a physical-cyber-social computing prism plays an important role in the human participation. As mentioned previously, sensor data are semantically annotated during the perception layer, supporting *sensor integration*, *sensor interoperability*, Semantic Stream Reasoning (SSR), Data Stream Mining (DSM), and Complex Event Processing (CEP). From a cognitive view, these *device-centric* (or *sensor-centric*) ontologies are overly technical or low-level to be interpreted and classified by end-users. This limits the capacity of human intervention in *data analytics* and *data fusion*, which minimizes the exposure of personal information but maximizes information accuracy and process automation. Therefore, the

13. https://www.mturk.com/mturk/welcome (accessed on 26/04/2017)
14. http://www.crowdsource.com/workforce/ (accessed on 26/04/2017)

high-level semantic abstraction provided by some *data analytics*, such as SSR, DSM, and CEP, can provide a more suitable and meaningful information layer to support human participation and, consequently, privacy harm perception. Nonetheless, *data analytics* are typically deployed on the application side, where privacy mechanisms are under *recipient sphere* influence and control. In this model, individuals who are concerned about their privacy do not have other choice but trust applications to access its personal information and sensor data to extract only non-private information.

INTENTION OF DATA USAGE At last, the *intention of data usage* concerns the usage of the collected data afterward. The concept of trust is intrinsically related to it and currently gravitates in the application layer, where business normally administrates information workflow. In particular, the IoT sensing service that continuously and opportunistically streams data from *IoT objects* is extremely sensitive. Once published and made available in the service layer, there is no guarantee that an application or service will not misuse this data stream to promote privacy attacks. For example, specific privacy attacks are developed to exploit stream queries to increase the level of confidence using some background knowledge. If intentions are specified for data usage, privacy preservation mechanisms could anticipate them in order to evaluate permission for data publishing.

2.4 ENABLING TECHNOLOGIES FOR PRIVACY PRESERVATION IN THE IOT

As described in the previous section, the complex Cloud-IoT infrastructure opens several aspects and issues related to privacy. The flow of sensor data and personal information in the IoT sensing can be exploited varyingly. In this section, we present the main enabling technologies that support the Sensing as a Service (S^2aaS) and the key concepts that we employ in our *privacy by design* solution and *privacy by policy* mechanism.

2.4.1 Analysis Criteria

Before discussing these works, we present the rationale used to elicit these enabling technologies. As mentioned in Section 2.2, efficient privacy strategies must address the inference intention of data processing techniques, the minimization of public data release, the data quality made available, the motive of data usage, and *data consumer*'s trust.

The first step toward efficient privacy preservation in the IoT (or in any informational context) is empowering individuals to choose the type of information they want to expose based on intelligible information, knowledge or wisdom, instead of sensor data.

The balance between *the right to seclusion* and *the opt-in choice for participatory sensing* demands privacy strategies with multiple mechanisms (*plurality*) that adapt with minimum human intervention (*contextuality*) and based on clear privacy policies (*contestability*).

For this reason, enabling technologies that can support the preventive evaluation of privacy policies based on the inference intention and human context. Therefore, the inference intention of data processing techniques should be represented explicitly somehow, allowing automatic interpretation and evaluation. The *in-network* data processing and privacy verification should take place into a controlled environment that meets the requirements to minimize a large amount of public data release commonly found in IoT streaming systems. Lastly, the motive of data usage and trust model should be encompassed in the access control model and design of the privacy verification. Therefore, we investigate technologies in the IoT sensing from a sensor and data-centric perspectives that deal with:

— In-cloud sensor virtualization which addresses the resource restriction of physical sensors, bringing the data processing and privacy verification to a *joint sphere* where it is possible to annotate semantically sensor data, and thus, assure sensor provenance;

— Data Stream Mining (DSM), Complex Event Processing (CEP), and Semantic Stream Reasoning (SSR) approaches that may infer any private personal information. Since these approaches produce semantic annotation along with its outputs, privacy verification can be incorporated along using Semantic Web technology *on-the-fly*. Additionally, these approaches perform KDDM processes that can be represented semantically and then evaluated to scrutinize *data consumer*'s motive and to prevent unintended inferences.

These technologies are investigated also from a privacy viewpoint. From a human aspect, we focus on *personal and household information ownership* of sensing services that have *humans as targets of sensing* in an *individual scale*, considering *semantic and cognitive perception*. The *human as targets of sensing* focuses on the personal information which is directly related to a person identity that can be extracted using data collected by A summary of our investigation scope is presented in Table 2.3.

SYSTEM PARTITIONING SCHEMES The approaches to the development of privacy enforcement *in-network* or *in-cloud virtualized* are particularly interesting due to the Cloud elasticity and the possibility to provide a neutral and safe environment to store and process personal information. The shift of *data processing* towards neutral independent parties in the *joint sphere* allows us to implement *privacy by design* away from the *data consumer*'s trust paradigm. By offering *data processing or storage* as a service, these independent parties can bill for privacy-aware S^2aaS by demand.

In [93], the data processing in large-scale sensor network was originally addressed using *virtual sensors* as part of a Global Sensor Net-

Perspective	Key concept	Issues & Characteristics
device-centric	system partitioning scheme	in-cloud virtualized
	sensor discovery and interoperability	sensor provenance
		privacy
data-centric	data streams	semantic stream reasoning
	data analytics	data stream mining
		complex event processing
human-centric	information ownership	personal and household
	participatory sensing	humans as targets of sensing
	opportunistic sensing	user awareness / human behavior
		ambient awareness / contextual environment
		social awareness / social context
	sensing scale	individual sensing
	participation in data analysis and data fusion	semantic perception and cognitive perception

Table 2.3 – Analysis criteria based on IoT perspectives

work (GSN) middleware. In this work, the *virtual sensors* are logical abstractions of one or more IoT object or other *virtual sensors* that capture, filter, and aggregate sensor data. The data stream processing mechanisms are natively implemented in GSN *virtual sensors* through SQL-like query language – TelegraphCQ[15]. Later in [27], the xGSN is proposed, leveraging the GSN capability using a semantics-based approach to address *sensor interoperability*, *sensor discovery*, Semantic Stream Reasoning (SSR) and *data analytics*. In xGSN, *virtual sensors* are extended and semantically represented using the SSN-O, which allows specifying sensor characteristics, platforms, observations, and sensing conditions. The xGSN *virtual sensor* provides wrappers that establish communication to the perception layer; and extensible data processing classes that allow local data filtering, aggregation, and processing. Each *virtual sensor* instance has an associated sensor instance in a triple cloud store. In addition, streaming observation annotations are generated in real-time and stored in a triple cloud store, such as the Linked Stream Middleware (LSM)[16], or redirected straight to a query processor, such as CQELS and EP-SPARQL. The LSM-based strategy allows asynchronous data analysis and high latency scenarios, while direct query processors must address scalability issues in order to deliver semantic streaming querying in (*quasi*) real-time. Regarding the abstraction level, the ontological framework provided by the SSN-O permits specifying sensor data semantic, such as geographic location points. However, while xGSN leverages sensing service by incorporating Semantic Web technology, SSN-O is not expressive to represent *virtual sensor* capacity and provenance, restricting *sensor discovery* and *sensor interoperability* capabilities. Also, the lack

15. `http://telegraph.cs.berkeley.edu/telegraphcq/v0.2/` (accessed on 26/04/2017)
16. `https://code.google.com/archive/p/deri-lsm/` (accessed on 26/04/2017)

of KDDM process representation limits the evaluation of inference intention of *virtual sensors*. Currently, xGSN *virtual sensors* provide a basic access control based on authorization and authentication, but no privacy preservation mechanism.

In [26], another work for *in-network* partitioning scheme is proposed based on multiple cloud tiers. Similarly, to the xGSN, this approach shifts from *a 'collect sensor data now and analyze it later' scenario to a usage scenario that directly provides meaningful information from in-network processing of sensor data*. The implementation of sensor integration and management services, such as *sensor discovery* and *sensor interoperability*, in Cloud-IoT infrastructures nearby physical sensors, can contribute to the network latency optimization. In this approach, location specific cloud agents, such as supercomputers, mobile devices, gateways, are made available to provide computational through conventional cloud platforms. These local cloud platforms constitute the first tier clouds that hide technical complexities and sensor heterogeneity in order to provide a unified and standard sensor interface and a distributed multi-tier infrastructure for *data analytics* using the concept of *virtual sensors*. Even though this approach draws attention to the benefits of *in-cloud* data processing, most of the works for privacy preservation based on this platform focus on security instead of privacy issues.

SENSOR DISCOVERY AND SENSOR INTEROPERABILITY The discovery and search of sensing resource in the S^2aaS is one of the most important functionalities in the IoT. As previously presented, the semantic annotation about observed features and properties provides means to discovery sensors and merge their data. In [94], an example of privacy enforcement is implemented in the *sensor discovery* and *sensor interoperability* services. An ontology-based trust model for Semantic Sensor Networks (SSNs) is introduced to represent trust relationships between *data consumers* (trustees) and *data producers* (trustors). The relationship is represented as a vector comprising both involved parties, the value that represents this relationship, trust type, scope of interest, and trust measurement function.

The type can be expressed as functional (direct between two agents), referral trust (transitive indirect), or non-functional trust (distrust). The scope captures the attribute on which the trust relationship is valid, such as some property of a feature of interest. The trust value is a discrete or categorical data used to evaluate trustworthiness, such as partial ordering or the binary representation, such as 0 and 1, false or true, trusted or non-trusted; and can be calculated according to a domain-specific function. Trust function represents the algorithm used to compute trust validation, aggregation, and management, such as policy-based, reputation-based, and evidence-based trust. These trust relationships are used to grant or deny access to sensor data in *sensor interoperability* and *sensor discovery* sensors.

However, for efficient privacy policy evaluation, one needs to understand what kind of personal information may be inferred from the sensor observation stream. Ontologies commonly adopted to this end, such as the SSN-O, do not support representing the input and the result of KDDM processes, such as those implemented by DSM, CEP, and SSR. Consequently, even if *sensor discovery* and *sensor interoperability* are layers that certainly intermediate applications and sensors (or services and sensors), they can only enact privacy enforcement based on raw physical sensor data or result of KDDM process executions. Neither solution is optimum because not the raw physical sensor data contains relevant personal information to be evaluated nor the latter prevent the execution of unintended KDDM processes.

DATA STREAMS The IoT sensing service relies mainly on the data stream and *on-the-fly* semantic enrichment to enable mainly *sensor discovery* and *sensor interoperability* services. Data Stream Management Systems (DSMSs) are specialized in dealing with large and transient data that is continuously updated, where no assumption can be made about data arrival order and boundness. DSMS isolates portions of the data stream in *windows*, transforming it into relational tables in order to use query mechanism from the relational model. These query mechanisms are issued once and run continuously and incrementally producing new output streams over time from one or more append-only input data streams. Similarly, to the relational model, two type of query algebra are defined for *data streams*: *stream-to-stream* and *mixed algebra* [92]. The former implements data operator that transform incrementally one or more streams into an output stream, such as the SQL statements for selection, projection, join, and aggregation functions. The latter includes three operator categories: *stream-to-relation operator*, *relation-to-relation operator*, and *relation-to-stream operators*. The first category consists of operators that transform streams into relation, such as a time-based sliding window or tuple-based sliding window. The second category refers to relational operators, such as the SQL statement DISTINCT. The third category consists of operators that transform relations back to the stream, such as Continuous Query Language (CQL)~[95] statement ISTREAM and DSTREAM, which insert and delete streams respectively. DSM lays the ground for more complex data processing over S^{2}aaS. However, the approach itself provides only a limited capacity to process data through its aggregation built-in functions.

SSRs use the same concepts to query semantically enriched data streams but incorporating reasoning capacity in the process. Observation stream annotation is commonly established in S^{2}aaS with SSN-O. Along with the Semantic Web technology, SSR can leverage the simple relational operations of *general data streaming processing* into higher-level semantic perception operations. In [70], the most representative work of the capacity of SSR is presented: an ontology-based framework to support intelligence data analysis of sensor data (IDA framework). The approach uses four ontologies to provide perception and classification of qualitative temporal patterns: Temporal Abstractions

Ontology (TAO) [70], SWRL Temporal Ontology (SWRLTO) [17], DOLCE-DnS UltraLite (DUL) [18] and SSN-O. The logical upper-level structure of DUL makes possible to aggregate available domain-specific knowledge based on the same upper ontology. The semantic enrichment incorporates sensor information, temporal information modeling, temporal abstract entities that enable stream reasoning capabilities using OWL and Semantic Web Rule Language (SWRL) technology to infer knowledge from S^{2}aaSs. Regarding privacy principles, the approach demonstrates the potential of Semantic Web technology to draw conclusions about higher-level information based on semantically annotated sensor data.

DATA ANALYTICS The integration of powerful *data analytics* in the S^{2}aaS combines the concept of *data stream mining* [67, 71] and the IoT sensing service. As previously mentioned, by moving the *data processing* model to the Cloud-IoT, privacy mechanisms can be prompted by anticipation based on the inference intention of KDDM process. In order to foresee unintended inference intention, we need to interpret KDDM process representation and outputs. Therefore, we discuss *data analytics* of data stream that produces interpretable results: Data Stream Mining (DSM), Semantic Stream Reasoning (SSR), and Complex Event Processing (CEP).

In [96], sensor data stream is segmented to infer activities using supervised vector machine techniques. These applications are important in the S^{2}aaS because it constitutes part of the data-to-decision path that can be shifted from the application layer to the *in-cloud virtualized* partition. In [97], the KDDM workflow is designed to be executed in parallel to the IoT sensing service. These approaches explore KDDM as a service and incorporated it into the IoT sensing service to provide knowledge or higher-level information to the application and business layer. These KDDM techniques output data generalizations, patterns, or other technical predictive model or statistics that normally demands expertise in mathematical models. In these cases, the privacy breach consists in the incorporation of KDDM techniques as part of the S^{2}aaS without proper interpretation, which could be active, for instance, using semantic annotation.

Sophisticated *data stream mining* approaches that use sensor data and DSMS query mechanisms have been proposed [98, 99, 100, 96, 101]. This type of works focuses on applying KDDM techniques to extract data generalizations from sensor data streams.

In *complex event processing*, the usage of KDDM techniques in sensor data streams or lower-level event streams(*primitive events*) focuses on making sense of higher-level information (*composite events*) [92], encapsulating this transformation process into a conceptual processing

17. https://github.com/protegeproject/swrlapi/wiki/SWRLAPITemporal (accessed on 26/04/2017). SWRL is a W3C query languange and protocol. Further information in Chapter 3.

18. http://ontologydesignpatterns.org/wiki/Ontology:DOLCE+DnS_Ultralite (accessed on 26/04/2017). DOLCE is an upper level ontology. Further information in Chapter 3.

block. The association of precise semantics to the information being processed and inferred allows that data consumer (application, business, and services) take actions based on this semantic annotation. On top of that, new streams of *composite events* deliver a higher-level information stream to applications instead of raw sensor data or *primitive event* stream. In [102], an open source system called ETALIS proposes to detect *complex events* over streaming data, following a deductive rule-based paradigm to evaluate domain knowledge *on-the-fly* and recognize pattern related to semantics. ETAILS implements a rule-based language for events, expressing single or multiple event occurrences, sequence, interval, precedence, and comparison. To enable semantic reasoning over complex event pattern, EP-SPARQL is developed to represent background knowledge about complex events and its relation to sensor data. ETALIS represent a class of application that leverages the stream reasoning, employing logic inference to infer higher-level information. This approach to abstract information using semantics can be incorporate in privacy mechanisms to decrease efficiently *quantity* of data, while increasing the expressiveness and context-awareness of semantic rule-based privacy policies. As the KDDM process in the IoT becomes more autonomous, SSR and CEP processes can be verified and interfered *on-the-fly*, facilitating the shifting of privacy enforcement to the *joint sphere* control.

2.5 CONCLUSION

In this chapter, we reviewed the key concepts of the Internet of Things (IoT) and its main enabling technologies. We discussed about IoT platforms and its convergence with the Cloud Computing that extends the IoT capacity. In addition, we pointed out main concepts of privacy engineering that will guide us in the rest of this manuscript to analyze related works and guide the definition of a *privacy by design* model. In order to efficiently scrutinize the IoT architecture and its design to propose a holistic *privacy by design* model, we present the main IoT technologies that are interrelated to provide IoT functionalities from three viewpoints: device-centric, data-centric, and human-centric. Lastly, we described in greater details some enabling IoT technologies related to the privacy paradigm that we intended to develop, delineating the scope of our investigation.

3 SENSOR WEB, META-MINING AND BEHAVIOR COMPUTING

CONTENTS

INTRODUCTION

The flexibility of the Semantic Web to represent knowledge using OWL can be employed to provide formal and user-friendly privacy mechanisms. Its formality permit to specify privacy policies, personal information and Privacy-Enhancing Technologies (PETs) unambiguously, as well as to infer information based on its reasoning applicabilities. In addition, Semantic Web technologies offer human-readable features that allow creating user-friendly interfaces to end-users. In this chapter, we review and investigate enabling technologies and approaches that will support to define our ontology for personal information and our *privacy by design* model.

In Section 3.1, we start by introducing the Semantic Web technologies and key concepts of the Resource Description Framework (RDF) framework, Ontology Web Language (OWL), ontology levels, and SPARQL Protocol and RDF Query Language (SPARQL). In Section 3.2, we describe the SSN-O, a *de facto* ontology for the Sensor Web, that is extensively used in the IoT services. We present a brief description of its origin and why SSN-O has become widely adopted. Moreover, we investigate its capacity to represent personal information and KDDM scenarios that are commonly found in the IoT and on which we based our *privacy by design* model. Next, in Section 3.3, we formally define the concept of Knowledge Discovery and Data Mining (KDDM) and review ontologies that propose its representation. In particular, we justify the reasons why we believe that the Ontology of Data Mining (OntoDM) can address SSN-O limitations that concern

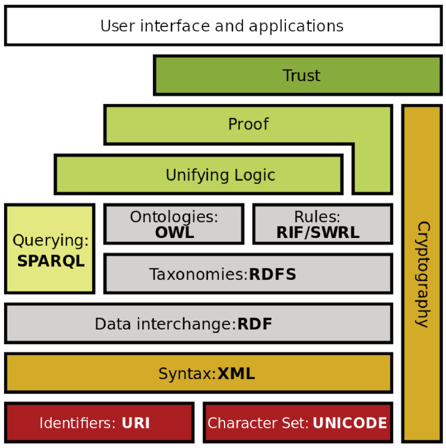

Figure 3.1 – Semantic Web Stack according to the W3C [1]

sensor provenance for *virtual sensors* and KDDM processes. In Section 3.4, we present key concepts of Behavior Computing (BC) that support the modeling and representation of personal information in behavioral contexts. Then, in Section 3.5, related works for representation of personal information using ontologies are discussed. Lastly, in Section 3.6, we conclude how these approaches and technologies can support the definition of our ontology for personal information on the Sensor Web.

3.1 SEMANTIC WEB TECHNOLOGY

The Semantic Web is a W3C effort to produce a formal knowledge representation to the WWW capable of expressing semantics for the information on the web. The W3C aims at standardizing concepts and encodings for the knowledge representation, shifting Web data towards a linked and semantic world, so called the Semantic Web. A conceptual and technological stack of the Semantic Web is illustrated in Figure 3.1. Its main technologies are described in the following subsections.

3.1.1 Resource Description Framework

The Resource Description Framework (RDF) is a knowledge modeling framework proposed originally to describe links between resources in the WWW. Instead of the WWW Uniform Resource Locators (URLs) referencing other URLs, RDF was meant to express information about the web content and its context. Differently, from URLs which are always related to a communication protocol, such as the Hypertext Transfer Protocol (HTTP) and File Transfer Protocol (FTP), RDF requires a more general resource representation. The concept of Uniform Resource Identifier (URI) was then extended from URL

1. https://en.wikipedia.org/wiki/File:Semantic-web-stack.png (accessed on 26/04/2017)

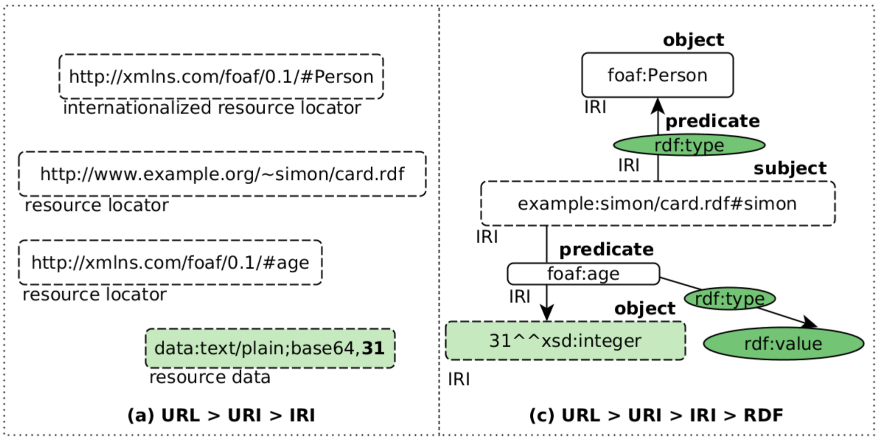

Figure 3.2 – Examples of URL, IRI, and RDF instances

to represent not just locators, but also identifiers, data about informational and physical resources, such as documents, links, people, physical objects, computers, and abstract concepts [103]. Since 2005, the new Internet standard Internationalized Resource Identifier (IRI) was introduced, aiming to extend URI to express things (instead of just resources) described using concepts defined in ontologies. An RDF statement is composed by a simple triple format:

`<subject> <predicate> <object>`

The *subject* and *object* represent the resources being related, while the *predicate* expresses their relationship. The concept of *subject* and *object* are extensible, based on IRI or self-denoting datatypes called *literals*. An example of RDF graph is presented below.

```
@prefix foaf:<http://xmlns.com/foaf/0.1/> .
@prefix example:<http://www.example.org/~simon/card.\ac{RDF}#>.

example:Simon   rdf:type  foaf:Person  .
example:Simon   foaf:age   <"31",xsd:integer> .
```

Figure 3.2 illustrates the evolution from the URL to RDF semantics using this RDF graph. The URL for a file (`http://www.example.org/~simon/card.rdf`) (accessed on 26/04/2017)) and for the elements *Person* and *age* in the Friends Of A Friend Ontology (FOAF) page [2] are presented in Figure 3.2.(a). These URLs can be referred, but no further semantics can be inferred from this relationship. On the other hand, IRI allows referencing any type of information. This semantic representation permits addressing *resources* (parts of information) to construct the notion of rich context. The triple format of RDF statements allows referring these resources, as depicted in Figure 3.2.(b). For example, the element 'Simon' in *example* [3] is of type (rdf: [4]:type) 'Person', which is defined in foaf [5]:Person. Furthermore, the element

2. `http://xmlns.com/foaf/.0.1/` (accessed on 26/04/2017)
3. **example** is a namespace for `http://www.example.org/~simon/card.rdf` (accessed on 26/04/2017)
4. **rdf** is a namespace for `https://www.w3.org/TR/rdf11-concepts/` (accessed on 26/04/2017)
5. **foaf** is a namespace for `http://xmlsns.com/foaf/0.1/` (accessed on 26/04/2017)

Constructor	N-triple	First-Order Logic
Class specialization	**C1** rdfs:subClassOf **C2**	$\forall(C1(X) \Rightarrow C2(X))$
Property specialization	**p1** rdfs:subPropertyOf **p2**	$\forall X \forall Y(p1(X,Y) \Rightarrow p2(X,Y))$
Restriction of property domain	**p** rdfs:domain **C**	$\forall X \forall Y(p(X,Y) \Rightarrow C(X))$
Restriction of property co-domain	**p** rdfs:range **C**	$\forall X \forall Y(p(X,Y) \Rightarrow C(Y))$

Table 3.1 – Main RDFS constructors

'Simon' can be related to the integer value '31' (resource data) through the predicate foaf:'age'.

3.1.2 RDF Schema and Ontology Web Language

In order to model and represent the semantics related to these resources used in the RDF, the W3C suggests the RDF Schema (RDFS) [104]. It extends the RDF semantics, providing axioms to describe groups of related resources and the meaning of the relationship between these resources. RDFS allows defining the class (*rdfs:Class*) of a resource, which property (*rdfs:Property*) a specific *predicate* represents, or which *datatype* a data value corresponds to. The main RDFS constructors are presented in Table 3.1 using the N-triple[6] encoding and the *First Order Logic* notation.

The OWL is defined on top of RDF and RDFS semantics, extending their expressiveness to describe knowledge about things, group of things, and relation between things [105]. An ontology is a set of logical descriptive statements about some domain of interest, composed of a set of three syntactic categories [106]:

— *Axioms*: statements expressed using the RDF representation and asserted as true in the scope of an ontology. For example, by defining the class Male as a specialization of the class Human using the ***subclass** axiom*, this statement is considered true when the ontology is used during a reasoning or inference;

— *Entities*: an atomic constituent of statements, such as objects, categories, and relations, identified by IRIs. Objects are called ***individuals***, categories denoted as ***class***, and relation referred as ***properties***. OWL accepts two type of ***individuals***: named (*owl:NamedIndividual*) and anonymous (represented by individuals which declarations starts with "_:" in RDF graphs). ***Class*** can be understood as a set of individuals, where *owl:Thing* referent the category of everything and *owl:Nothing* the empty set of individual. ***Properties*** are specialized in three types: *object properties* (*owl:ObjectProperty*) that express object-to-object relation, *datatype properties* (*owl:DatatypeProperty*), which assign data values directly to objects, and *annotation properties* (*owl:AnnotationProperty*) that assign data values as annotation to objects.

6. https://www.w3.org/TR/n-triples (accessed on 26/04/2017)

— *Expressions* represent complex representations constructed based on combinations of up-mentioned entities or expressions.

The OWL expressiveness relies on its semantics to represent the relation between classes, their properties, and relationships to datatypes. For example, class can be associated based on equivalence (*owl:equivalentClasses*), disjointedness (*owl:disjointWith*), union (*owl:disjointUnionOf*), intersection (*owl:intersectionOf*) , complement (*owl:complementOf*), and so forth. Properties can be defined according to functionality (*owl:FunctionalProperty*), as inverse of another property (*owl:inverseOf*), reflexive (*owl:ReflexiveProperty*), symmetric (*owl:SymmetricProperty*), transitive (*owl:TransitiveProperty*), cardinality (*owl:minCardinality*, *owl:maxCardinality*, *owl:cardinality*), and so on. These attributes are defined along with its logical negatives, such as asymmetry and symmetry properties.

Two important features of OWL are the universal and existential operators that allow restricting properties according to values, similarly to the *First Order Logic* operators. In Listing 3.1, some of these axioms are depicted in a fragment of the FOAF.

Listing 3.1 – Fragment of FOAF ontology

```
<rdf:RDF xmlns:foaf="http://xmlns.com/foaf/0.1/"
         xmlns:owl="http://www.w3.org/2002/07/owl#"
         xmlns:wgs84="http://www.w3.org/2003/01/geo/wgs84_pos#"
  <owl:Ontology rdf:about="http://xmlns.com/foaf/0.1/" />
  <rdfs:Class rdf:about="foaf:Person" rdfs:label="Person" >
    <rdf:type rdf:resource="owl:Class" />
    <owl:equivalentClass rdf:resource="http://schema.org/Person"
        />
    <rdfs:subClassOf>
          <owl:Class rdf:about="foaf:Agent"/>
    </rdfs:subClassOf>
    <owl:disjointWith rdf:resource="foaf:Organization"/>
    <owl:disjointWith rdf:resource="foaf:Project"/>
  </rdfs:Class>
  ...
```

OWL also defines semantics to compare individuals, such as *owl:sameAs*, and *owl:differenteFrom*. In term of ontology properties, OWL permits importing and configuring ontology versioning (*owl:versionInfo*, *owl:priorVersion*, *owl:backwardCompatibleWith*, *owl:DeprecatedClass*). In Figure 3.3, the concepts of foaf:Person and foaf:age are defined as OWL class (*owl:Class*) and extended using the properties *rdfs:subClassOf* and *owl:equivalentClass*. OWL semantics make it possible to conceive a conceptualization of the world to describe resources (IRIs) that can be typified (using RDFS semantics). Additionally, restrictions to the cardinality of this typification, such as *owl:maxCardinality*, leverages the level of expressiveness of RDFS. Lastly, specific semantics for annotation, such as *owl:label*, transform ontology in *representational artifacts*, allowing computational agents to reason logically using the OWL semantic and present intelligible results for end-users using entity annotations.

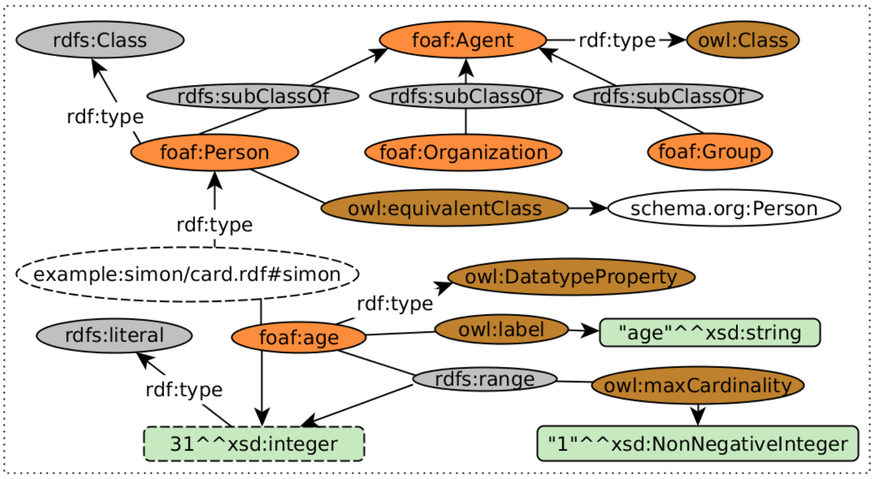

Figure 3.3 – An example of RDFS and fragment of FOAF ontology

3.1.3 SPARQL Query Language and Protocol

The OWL, RDFS, and RDF semantics provide support for logical inferences. W3C recommends the SPARQL to express reasoning and update queries for RDF graphs and implement endpoint interfaces. SPARQL query language is an SQL-like language that uses RDF graph criteria matching in an *open-world*[7] assumption [107]. *Where* clause is then specified using the RDF format, i. e. a <*subject predicate object*> triple, separated by the *and* operator denoted by a "."(dot). The use of prefixes (*prefix*) is commonly employed in SPARQL queries for simplification, as shown below:

```
PREFIX foaf:<http://xmlns.com/foaf/0.1/>
SELECT ?x ?age
WHERE
{ ?x rdf:type rdf:class.
  ?x foaf:age ?age }
```

This SPARQL query retrieves the following result if applied to the example of RDF depicted in Figure 3.3:

class	age
foaf:Person	"'31'^^xsd:integer"

Besides the *select* clause, SPARQL has also *construct*, *ask*, and *describe* as query forms. *Construct* returns an RDF graph with the query result. *Ask* returns true or false if there are matches for a specific query pattern. *Describe* returns a single result RDF graph containing all resources linked to a specific resource (IRI). SQL-base clauses, such as *order by*, *limit*, *offset*, *distinct*, *join*, *left join*, *union* and so forth are also available in SPARQL query language. The *where* clause provides the representation of graph patterns to be retrieved. On the other hand, the logical operator to test values in SPARQL is implemented using the clause *filter*.

7. Oppositely to the *close-world* assumption on which relational database paradigm is based and where facts are considered false if it is not present in the database.

Syntax	Description
iri	A IRI, a path of length one
ˆpp	Inverse path
!iri	Negated path
(pp)	Grouped path where the brackets control precedences
pp_1 / pp_2	A sequence path (pp_1 and pp_2)
pp_1 | pp_2	A alternative path (pp_1 or pp_2)
pp*	A path of zero or more occurrences of pp
pp+	A path of one or more occurrences of pp
pp?	A path of zero or one occurrences of pp
pp{n,m}	A path of length between n and m of pp
pp{n}	A path of exactly n occurences of pp
pp{,n}	A path of exactly n occurences of pp
pp{,m}	A path of exactly n occurences of pp

Table 3.2 – SPARQL property path syntax

Another important feature of the SPARQL language is the concept of *property path*. A property path expression (or simply a path) is similar to a string regular expression that uses object property instead of characters [108]. It represents a possible route of arbitrary length through a graph between two graph nodes. In this case, query evaluation determines the matches for a path expression. Table 3.2 presents the main syntax forms of SPARQL property paths, where *iri* expresses IRI, and *pp* property path.

Good practices in ontology engineering suggest to separate terminology definition – Terminological Components (TBox) – from individual assertion definitions – Assertion Components (ABox) [109]. The first represents the elements of an ontology, i.e., classes and properties, which is structural and *intensional*[8]. The latter corresponds to attributes of individuals, the roles between instances, data values, and class membership (*rdf:type*). OWL reasoning is commonly split using this TBox/ABox concept, facilitating the analysis and implementation of logic-based reasoning tasks. TBox reasoning is related to property path inference, classes and properties equivalence, class satisfiability (verification whether a concept is consistent, i. e. different from *owl:Nothing*), logical implication (rule-based), and so on. ABox reasoning observes the entailment between class and individual, can be used for knowledge base consistency (verification whether the ontology admits a minimum set of individuals), individual realization (find the best class match for an individual), individual retrieval etc.

3.1.4 OWL Profiles

The OWL offers profiles – commonly called fragments or sublanguages – that restrict OWL expressiveness in exchange of reasoning efficiency and decidability. Originally, OWL Lite and OWL Description Logic (OWLDL) were proposed as fragments of the OWL Full[9]. The

8. *Intension* is a linguistic, logic, and phylosophy property or quality connoted by a linguistic entity (word, phrase, symbol).

9. Full OWL semantic set

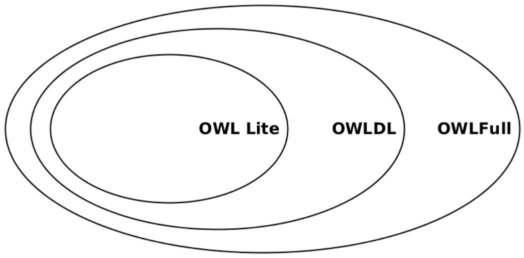

Figure 3.4 – OWL Profiles

OWLDL is a sublanguage of OWL that restricts a number of OWL constructs. It is related to (and based on) the semantics of the Description Logic (DL) *SROIQ* [110] and guarantees *decidability* in reasoning. The OWL Lite constitutes the same restrictions imposed by OWLDL, forbidding additionally constructs of unions, intersections, disjoints, data ranges, and data values.

Figure 3.4 illustrates the scope of each up-mentioned OWL profile. The second version of OWL (OWL2) defines the following profiles [111]:

— OWL2 EL profile: stands for basic reasoning problems in ontologies with very large number of classes and properties, such as those found in genealogy where ontologies can easily reach more than 25 thousand classes. It represents the smallest set of OWL expressiveness including only conjunction and existential restrictions.

— OWL2 RL profile: provides scalable reasoning and considerable expressiveness through implication (if/then) rules. It is typically adopted for large datasets that need to represent the existing data using (business) rules;

— OWL2 QL profile: aims at applications that use a very large volume of individuals, benefiting from the polynomial performance for query answering by restricting implications This profile can be translated into relational queries or UML class diagrams, for example.

3.1.5 Ontology levels

The conceptualization of a knowledge domain through ontologies requires common accordance about vocabulary and conceptual modeling designs. Upper-level ontologies and middle-level ontologies address this problem by defining ontology design choices and a common knowledge base for building new ontologies. The assumption behind upper-level ontologies is that, by generalizing ontologies from different domains, it is possible to extract a minimum set of classes and properties that represent these specific domain entities in a higher-level. Good practices in ontology engineering suggest the use of upper-level ontologies in order to align ontology entities in different levels of abstractions and to support designers to negotiate meaning. As illustrated in Figure 3.5, ontologies are

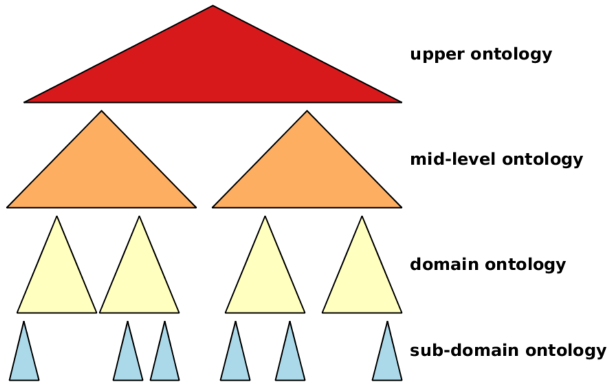

Figure 3.5 – Ontology level hierarchy

typically classified in four levels of abstraction. *Upper-level ontologies* (or *foundational ontologies*) map the reality according to specific ontological design choices, such as descriptive vs. revisionary, or endurantism vs. perdurantism. Several upper-level ontologies are proposed, such as Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) [112], Basic Formal Ontology (BFO) [113], and Suggested Upper Merged Ontology (SUMO) [114]. *Middle-level ontologies* correspond to domain-spanning knowledge, serving to bridge between general entities in upper-level ontologies and the domain-level ontologies. The Information Artifact Ontology (IAO), for example, is a middle-level ontology which attempts to capture the essence of information entities, the relationship between these entities, and their meanings [115]. *Domain-level ontologies* represent a specific domain of interest, generalizing categories of individuals. For instance, the SSN-O [40] proposes to describe sensors and observations. *Sub-domain-level ontologies* refer to particular sub-domain that is too specific to be considered a domain of interest, such as an ontology that specializes a specific type of sensor by extending the SSN-O.

3.1.6 Semantic Web Rule Language

Nonetheless, neither OWL nor SPARQL can express dynamic relations between individuals with which an individual has relations. For this purpose, the W3C recommends the SWRL to express axiom rules. SWRL is based on the combination of OWLDL and the Rule Markup Language (RuleML) [116] that extends the OWL syntax to represent implications between *antecedent* and *consequent*. For example, the statement *child of divorced parents*, which is not possible to be declared using OWL, can be expressed using SWRL. The rule **hasParent**(?x1,?x2) $\wedge$ **hasParent**(?x1,?x3) $\wedge$ **isDivorcedTo**(?x2,?x3) $\Rightarrow$ **ChildOfDi-**

vorcedParents(?x1) is presented in Listing 3.2 using SWRL syntax.

Listing 3.2 – SWRL example

```
Implies(
        Antecedent(hasParent(I-variable(x1) I-variable(x2))
                   hasParent(I-variable(x1) I-variable(x3)))
        Consequent(ChildOfDivorcedParents(I-variable(x1)))
)
```

3.2 SEMANTIC SENSOR NETWORK

Part of the Linked Data Web in the WWW comes from the network of connected sensors. The so-called *Sensor Web* represents a sensor network with integration and management capabilities that incorporate Semantic Web technology.

Aiming to propose a generic solution for the Sensor Web, the Open Geospatial Consortium (OGC) proposes a set of open standards for exploiting connected sensors and sensory system of all types called Sensor Web Enablement (SWE) [117]. The models, encoding, and services introduced by the SWE initiative have already been applied to several other standards, such as Observations and Measurements Schema (O&M) [118] and Sensor Model Language (SensorML) [119]. The focus of this initiative is to enable the discovery, determination of capabilities and measurement quality, access to parameters, retrieval of observations and coverages, task management, alert subscription, and publishing of sensors.

These standards provide syntactic interoperability, but no domain semantic compatibility [40]. In this context, ontologies and semantic technologies play an important role in sensor networks, enabling representation of domain knowledge related to a feature of interest and observations. In [120], an extensive analysis of aspects of sensors covered by sensor network ontologies is presented, classifying these aspects into four categories: sensor, physical, observation, and domain. However, as demonstrated by the authors, none of the surveyed ontologies met modern requirements of the sensor network and the Sensor Web. Based on that, the W3C Semantic Sensor Network Incubator group [10] proposed the Semantic Sensor Network Ontology (SSN-O) aiming to address those gaps related to sensors, observations, and domain knowledge representations.

SSN-O has become a *de facto* standard for the Sensor Web and operates at an abstraction level above technical details of format and integration, working with features of interest, observation, sensor and restrictions on quality [40]. The ontology is designed using the OWLDL profile of OWL2 and is aligned with the upper ontology DOLCE-DnS UltraLite (DUL) , a lightweight version of the foundational ontology DOLCE. DUL has a clear *cognitive bias* in the sense that it aims at cap-

10. http://www.w3.org/2005/Incubator/ssn/XGR-ssn-20110628/ (accessed on 26/04/2017)

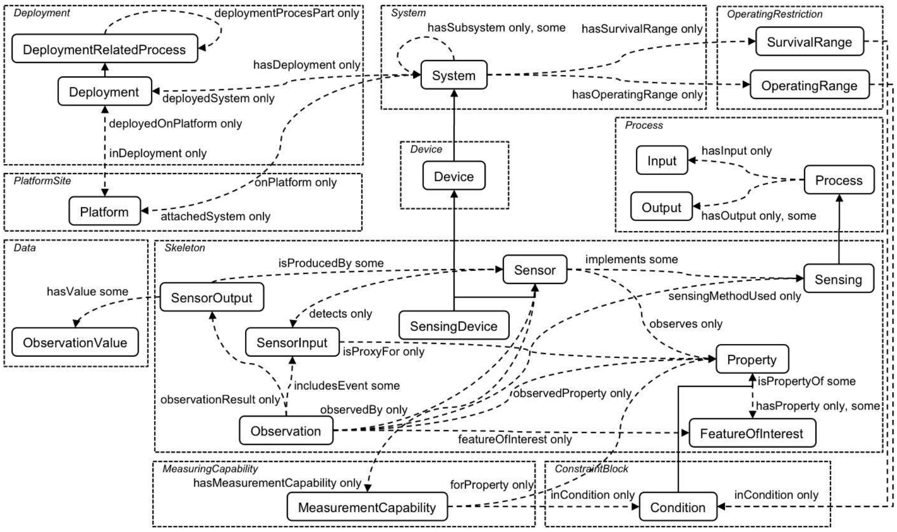

Figure 3.6 – SSN-O modules [40]

turing the ontological categories underlying natural language and human commonsense [112]. By aligning an ontology to SSN-O and DUL, an extensible *ontological framework* is provided, being compatible with other DUL-based ontologies and inheriting semantic matching that exists between DUL and other upper-level ontologies, such as the BFO [11].

The DUL conceptual framework distinguishes *abstract*, *information entities*, *objects*, *quality*, and *events*. This structure describes entities in a *posthoc* way, reflecting more or less the surface structure of languages and cognition [112]. An *abstract* entity represents anything that cannot be located in space-time, such as mathematical entities, format semantic elements, regions (*quality spaces*), and so forth. *Information entity* represents datum or information that can be concretely realized or not, such a document file (*information object*) or a printed document file (*information realization*). *Objects* in DUL are specialized into *physical objects*, *agents*, and *social objects*. The first represents anything that is located in a space region and has an associated mass. The second represents any agentive object that can be physical (e.g., animal, robot, person) or social (eg., corporation, community, collective). The third is a special entity in DUL that exists only within some communication event, in which at least one *physical object* participates in. *Social objects* exist for the sake of communication, expressed by *information object*, for: i) incorporating individuals (*social agents*, *places*); ii) collecting entities (*collection*); and iii) describing them (*description*, *concept*).

The SSN-O is organized conceptually in ten modules, as illustrated in Figure 3.6. The SSN-O class sensor is a DUL *physical object* that implements a DUL sensing process that observes a *property* of a *feature of interest*. In order to conceptualize the events that link observations and sensors, the Stimulus-Sensor-Observation (SSO) design pattern is present in the SSN-O [121]. The purpose of SSO design pattern is to bridge sensors and observations by focusing on *stimuli* as objects of

11. https://github.com/BFO-ontology/BFO (accessed on 26/04/2017)

sensing, which consequently constitutes a proxy for observed *properties*. Basically, SSN-O sensors are event-based (*stimuli*), and, consequently, work in conditions and constraints (e.g. limited power availability, variable data quality, limited memory, bad environmental conditions) that needs to be accounted in the post-processing phases. As implementation of sensing process, SSN-O sensors produce *sensor outputs* by detecting (only) *stimulus*. These *sensor outputs* have values of the type *observation value*, which are expressed by DUL regions, along with spatiotemporal or amounts, that can be extended using an external ontology, including, for instance, units of measurements. SSN-O is intended to be used as sub-domain ontology, having its classes specialized to represent specific sensors, features of interest (and their properties), and its sensing processes.

In addition, the SSN-O sensor can be represented in terms of its properties and technical specifications, such as *measurement capabilities*. SSN-O sensor is DUL systems that can be deployed on DUL *platforms*. *Systems* can be represented in terms of *survival range* and *operating range*, which are defined based on *conditions* (constraint block); and *platforms* in terms of its *deployment related process*. Thus, *provenance* and diagnosis of observations are inherent benefits of SSN-O, allowing evaluating the context in which data has been produced. Moreover, SSN-O takes a liberally inclusive view of what a sensor is: *any (**physical**) thing that observes*; allowing such sensors to be described as simple physical objects that play a role of sensing, as well as semantically enriched sensor systems that are described in terms of their components and methods of operation to support data interoperability and Sensor Web integration and management [40].

3.3 KDDM AND THE META-MINING

Virtual sensors are suitable to constitute a new conceptual layer that implements Semantic Stream Reasoning (SSR), Data Stream Mining (DSM), and Complex Event Processing (CEP) approaches. However, the main works for semantic sensor annotation do not cover representation for KDDM techniques which are implemented by these approaches. The SSN-O ontology defines sensor as physical entities and, although it is possible to represent the sensing process of these physical sensors, it does not provide representation for algorithms used in this process.

3.3.1 KDDM process

The concept of Knowledge Discovery in Databases (KDD) was originally defined in [122] as a "nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data". According to the authors, data mining is only one step in the KDD process focused on searching patterns of interest in data sets. Besides that, related fields, such as machine learning and logical rea-

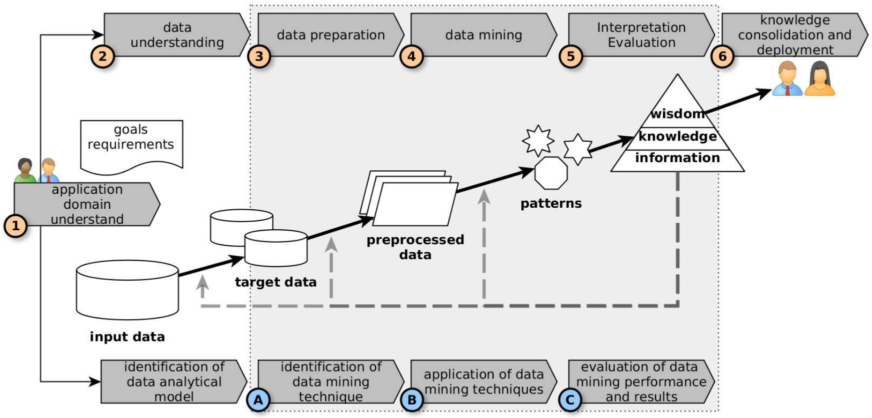

Figure 3.7 – Overview of the KDDM process

soning, were also included as part of this process, consisting of alternative techniques to extract knowledge. More recently, an attempt to describe knowledge extraction and involved technology in the knowledge extraction process has been described in the Knowledge Discovery and Data Mining (KDDM) model [123]. In this survey, several KDDM models are compared to each other, resulting in six general steps, as depicted in Figure 3.7, and described as following:

1. *application domain understand*: incorporates tools to define KDDM objectives and analysis requirements based on business problems, goals, and context;
2. *data understanding*: consists of tools for identification of data quality issues, data exploration, selection of data subsets, and usefulness of data;
3. *data preparation*: corresponds to tools for the preparation of dataset that will be used as input for data mining. This preparation includes methods for attribute selection, cleaning, outlier removal, functions to compose new attributes, data transformation, dimensionality reduction, and mechanisms to treat heterogeneity issues;
4. *data mining*: refers to the data mining, machine learning and reasoning techniques and their applications using the prepared dataset and generating generalizations, information, and knowledge;
5. *evaluation*: consists of tools for interpretation, filtering, validation, and visualization of the generated results based on the KDDM objectives;
6. *knowledge consolidation and deployment*: represents the incorporation of the discovered knowledge into a system or reports for end-user presentation.

We illustrate in the bottom of Figure 3.7 the steps of the KDDM that are automatable: A) *identification of data analytical model*, B) *identification of data mining techniques*, and C) *application of data mining techniques*. Each automatable step is aligned with its respective KDDM correspondent. In this manuscript, we focus on steps 3 to 5, as highlighted in

Figure 3.7. In particular, we aim to map these steps in *virtual sensors* in order to support IoT services that rely on steps A, B, and C. In addition, the remainder of this thesis considers KDDM and data mining as interchangeable concepts.

3.3.2 Meta-Mining

The MM research field consists of an extension of traditional *meta-learning* research domain. Meta-learning refers to the application of machine learning techniques to meta-data that describes past learning experiences in order to improve future performances and results of these algorithms [124]. Meta-mining extends the meta-learning approach to the full KDDM process. In this context, several ontologies are under development, aiming to model KDDM algorithms and process [125]. These ontologies have been employed to support the automation of KDDM composition and the optimization of KDDM algorithm execution by adjusting pre/post-processing steps based on meta-data and performance evaluation. None of these ontologies has been universally established. However, some criteria ground the decision for a most suitable ontology to describe KDDM techniques in the context of *virtual sensors*.

Most of these ontologies are complementary or overlapping. For instance, in [126] the Data Mining OPtimiation Ontology (DMOP) [12] is proposed to support all decision-making steps that determine the outcome of the data mining process in the data preparation, KDDM modeling, and KDDM evaluation [126]. DMOP is a OWL2 DL ontology that provides a conceptual framework to represent data mining tasks, algorithms, models, datasets, workflows, and experiments. In [127], an OWL2 modular OntoDM [13] is proposed. In this approach, core data mining entities are represented, such as datatypes, data sets, data mining tasks, data mining algorithms, functions, generalizations, workflow, and evaluation scenarios and so forth. OntoDM is based on the foundational ontology BFO and reuses other middle-level ontologies to represent auxiliary entities, such as algorithm executions, data items, and objectives. In [128], Exposé Ontology (Exposé), a generic ontology for data mining and machine learning representation of scientific experiment is proposed. The initial goal of Exposé was to represent experimental methodology based on data processing operations of the upper-level ontology SUMO. It provides a representation for experiment context, evaluation metrics, performance metrics, data sets, algorithms, which are complementary to those entities defined in OntoDM.

In this thesis, OntoDM were chosen to represent KDDM techniques that are implemented in *virtual sensors* based on its generality, that maximizes its adoption in different domains; its alignment to a foundational ontology, BFO version 1.1, which is compatible to DOLCE and consequently to SSN-O; and its design process that followed good prac-

12. DMOP: `http://www.dmo-foundry.org/DMOP` (accessed on 26/04/2017)
13. `http://www.ontodm.com/` (accessed on 26/04/2017)

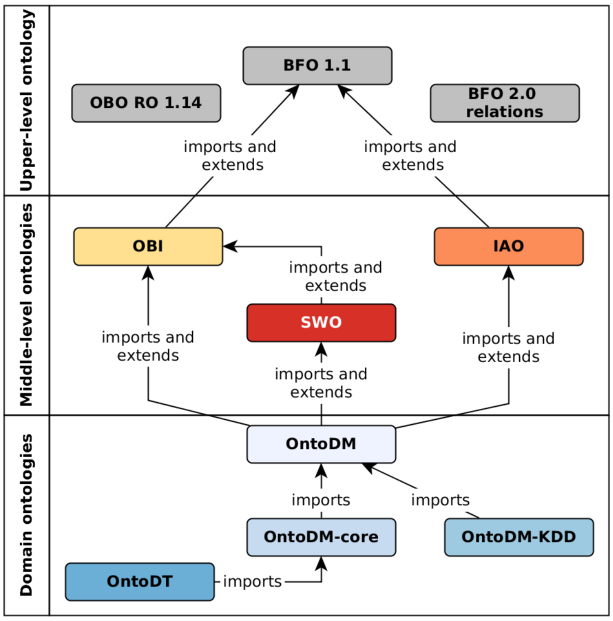

Figure 3.8 – OntoDM structure [129]

tices for ontology development by reusing class and relation from established upper-level and middle-level ontologies, as illustrated in Figure 3.8. The reconciliation between the BFO *realistic* and the DUL *cognitive* perspectives have been studied in [130], suggesting the most classes of both ontologies are compatible, which expands OntoDM interoperability.

3.3.2.1 *OntoDM*

The semantics provided by OntoDM consists of a family of three ontologies: OntoDM-core [131], Ontology for Data type (OntoDT) [132], and OntoDM for KDD (OntoDM-KDD) [133]. OntoDM-core is designed following Open Biomedical Ontologies (OBO) [14] Foundry design principles, such as the use of an upper-level ontology, formal ontology for relations, reuse of established ontologies, and preference for more strict ontology profiles. Similarly to the best practices for ontology engineering, these principles enable the interoperability of ontologies that adopt the same principles and reasoning among ontologies based on the same upper-level ontologies. Besides the foundational ontology BFO version 1.1, the ontological framework of OntoDM is composed by OBO Relational Ontology (OBO RO) version 1.14, BFO version 2.0 (relations), Ontology for Biomedical Investigation (OBI) [15] RC1 (release candidate 1) version, IAO [16], and Software Ontology (SWO) [17].

The expressiveness and ability to describe different use case of the OntoDM are due to the reuse of these ontologies that provide concepts

14. OBO Foundry: http://www.obofoundry.org/principles/fp-000-summary.html (accessed on 26/04/2017)
15. OBI: http://purl.obolibrary.org/obo/obi (accessed on 26/04/2017)
16. IAO: https://github.com/information-artifact-ontology/IAO/ (accessed on 26/04/2017)
17. SWO: http://theswo.sourceforge.net/ (accessed on 26/04/2017)

to describe the data mining classes, data transformation classes, and auxiliary data processing concepts. The OntoDM-core defines the core set of data mining entities, such as data mining objective, dataset specification, and algorithm implementation parameters. OntoDT defines basic entities to represent datatypes, such as properties of a datatype, its specification, or taxonomy. The OntoDM-KDD specifies basic entities to represent information discovery process and its components, such as knowledge discovery phases, workflows, and information discovery scenarios. For the sake of readability and space, we present a more detailed description of the OntoDM ontological framework in Appendix 9. In the remainder of this manuscript, OntoDM is used as the set of these three ontologies, except cases when we explicitly refer to them uniquely.

OntoDM is structured in three abstraction levels:

— *Specification level*: consists of classes to represent *dependent continuants*, such as data, datasets, data processing goal, data mining algorithm, and generalizations;
— *Implementation level*: refers to classes that represent *specifically dependent continuants*, such as the implementation of data processing algorithms, data mining algorithms, workflow, training data sets (roles), functions, operators that realize some implementation, parameters, and quality aspects;
— *Application level*: consists of classes to represent *planned processes*, such the execution of data mining algorithms and generalizations.

Theses layers are important to represent separately the specification of entities that participate in the KDDM processes, their implementations, and their execution. This three-level structure matches to our need of specifying these level of abstractions in *virtual sensors* in order to allows reasoning of inference intention using design specification (objective, process, type of algorithms, datatypes, and generalizations), deployment level (virtual sensors, algorithm implementations, functions, operators), and application level (virtual sensor executions, algorithm executions, datasets). The three-level structure provides flexibility in terms of representation of data processing operations and their specifications. In the following subsections, these levels are described briefly, restraining on classes relevant to the scope of this manuscript. Further information can be found in `www.ontodm.com` (accessed on 26/04/2017) or in the literature [129].

SPECIFICATION LEVEL The *specification level* of OntoDM-core consists of classes specializations of IAO Information Content Entity (ICE), OBI *data representational model* and OBI *protocol*. As depicted in dash-dotted boxes in Figure 3.9, OntoDM specifies specializations for IAO *data item* and Directive Information Entity (DIE), as well as for ICE itself. For *data item*, OntoDM defines *data example*, *parameter setting*, and specializes *data sets*. *Data example* represents one unit of data and it is part of a *DM-dataset*. *DM-datasets* are datasets formed by *data examples*, which can be specified according to a *data specification*, such as *bags* or *folds*. A *data specification* is a *specification entity* that **isAbout** a *data*

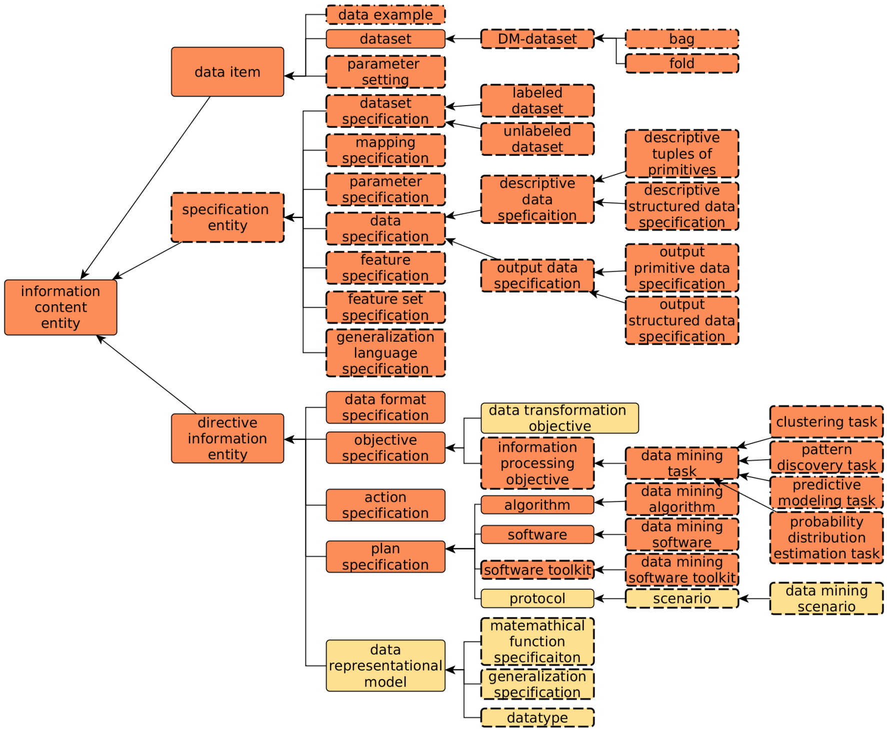

Figure 3.9 – OntoDM-core specification level. Orange boxes represent IAO classes. Yellow boxes represent OBI classes. Dash-dotted boxes represent OntoDM extension.

example. *Parameter setting* is a quality specification of some algorithm parameter.

A *specification entity* is related to other ICE entities using the OBO RO *is-about* relation, and it is specilized in *data specification*, *dataset specification*, *mapping specification*, *parameter specification*, *feature specification*, *feature set specification*, *generalization language specification*. *Data specification* can be of the type *descriptive data specification* and *output data specification*. The former consists of a specification for the datatype of the descriptive part of a dataset, while the latter denotes the datatype for data on the output part of a dataset, typically found in modeling tasks. These *data specifications* can be formed by primitives (string, integer, complex, discrete) or structured datatypes (tree, array, tuple). OntoDM uses the *mapping specification* to associate which part of the *data example* a concrete datatype applies to. *Parameter specification* consists of an algorithm implementation quality specification (*parameters*). *Feature specification* and *feature set specification* are specification entities related to primitive data feature or a set of data features (i. e., tuples) used to identify features for data mining algorithms. *Generalization language specification* consists in the language formalism used to express a *generalization* [18], such as trees, rules, neural networks, etc.

The DIE represents specifications of entities that can be concretized or realized, such as processes, functions, datatypes, and information processing tasks. *Data format specification* is a DIE that specifies the

18. Generalization is the outcome of applying a data mining task for a given dataset.

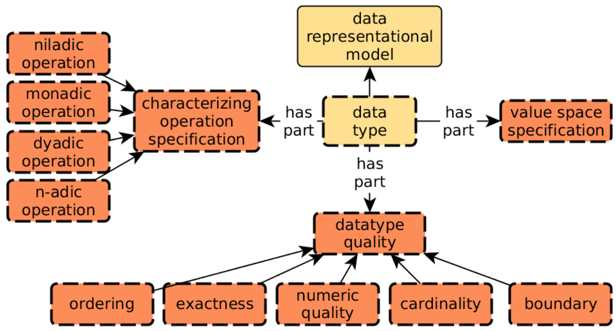

Figure 3.10 – OntoDT datatype. Yellow boxes represent OBI classes. Orange dash-dotted boxes represent OntoDM extension of IAO classes.

format a data is encoded, such as XML, RDF, and binary format. *Objective specification* is extended by the OBI as a *data transformation objective*, such as a normalization or a partitioning objective. OntoDM proposes to specialize the concept of *objective specification* in *information processing objectives*. Instead of data transformation, *information processing objective* represents algorithms to extract information from data using *data mining tasks*, such as *clustering task*, *pattern discovery task*, *predictive modeling task*, and *probability distribution estimation task*. In order to represent their *plan specification*, OntoDM extends the IAO classes *algorithm*, *software*, and *protocol* to represent *data mining algorithms*, scenarios (KDDM processes), and *data mining scenarios* (KDD processes) respectively. In addition, OntoDM proposes to represent *software toolkits* and *data mining software toolkits*, such as the Waikato Environment for Knowledge Analysis (WEKA) [19].

OBI extends DIE by specifying *data representation models*. From this concept, OntoDM specializes *mathematical function specifications*, *datatypes*, and *generalization specifications*. The *mathematical function specification* represents distance function sets, scoring functions, cost functions, optimization functions, and so forth. *Generalization specification* specifies the type of generalization, the datatype used to produce the generalization, and the generalization language.

19. WEKA:http://www.cs.waikato.ac.nz/ml/weka/ (accessed on 26/04/2017)

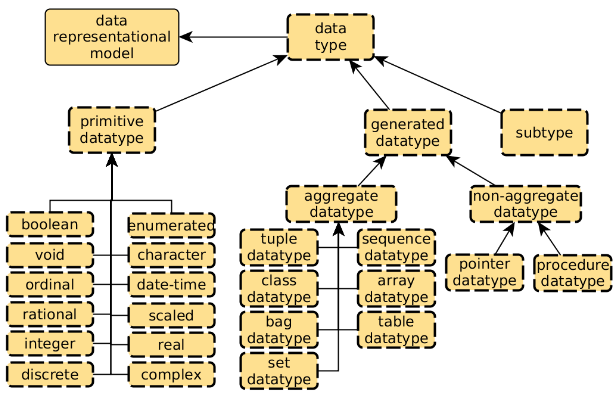

Figure 3.11 – OntoDT datatype taxonomy. Yellow boxes represent OBI classes. Dash-dotted boxes represent OntoDM extension.

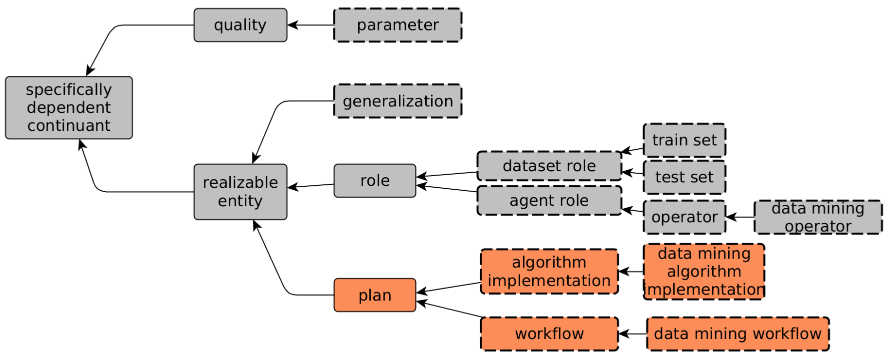

Figure 3.12 – OntoDM implementation level. Gray boxes represent BFO classes. Orange boxes represent IAO classes. Dash-dotted boxes represent OntoDM extension.

Datatype is defined in OntoDT and specifies a set of distinct values, their properties, and permitted operations, as illustrated in Figure 3.10. *Value space specification* is a *specification entity* that defines the collection of values. *Characterizing operation specification* is a *specification entity* that comprises types of operations (niladic, monodic, dyadic, *n*-adic) and their association to *value space specifications*. *Data quality* is a *specification entity* to represent *ordering*, *exactness*, *numeric quality*, *cardinality*, and the *boundary* of a *datatype*. *Datatype* is specialized in three subclasses: *primitive datatype*, *generated datatype*, *subtype*, as presented in Figure 3.11.

Generated datatypes can specify structured collections of *primitive datatype*, allowing to represent sophisticated data structures, such as *tuple*, *class* (similar to object oriented classes), *bag*, *set*, *sequence*, *array*, *table*, *pointer* (such as variable address reference), and *procedure* (an operation on values of other datatypes). *Subtype* represents restrictions on primitive or generated datatypes, such as a range of a discrete datatype.

IMPLEMENTATION LEVEL The implementation level of OntoDM-core consists of classes that are extension of the BFO *specifically dependent continuant* and IAO *plan*, as illustrated in Figure 3.12. OntoDM specializes BFO class *quality*, representing *parameters* for algorithm implementations; and BFO class *realizable entity*, specifying the concept of *generalization*. *Parameters* are *realizable entity*, being concretized by the implementation of an algorithm. *Generalization* has a dual nature and denotes the outcome of applying a *data mining task* (related to an algorithm execution through an **isSpecifiedOutputOf** relation), while still being a *realizable entity*. It acts as a representation of *data mining algorithm execution* outputs, and as a *generalization execution* that has an input *dataset* and an output *dataset*. For example, by executing a *predictive modeling task* over a given *DM-dataset*, a predictive model is outputted. Subsequently, by executing this predictive model using an input *dataset*, a predicted data set is produced.

BFO role entity is specialized in OntoDM to represent *dataset roles*, such as *train set* or *test set*, and *agent roles*. In addition, *algorithm implementation* is extended from the BFO *plan* entity, as an intermediary

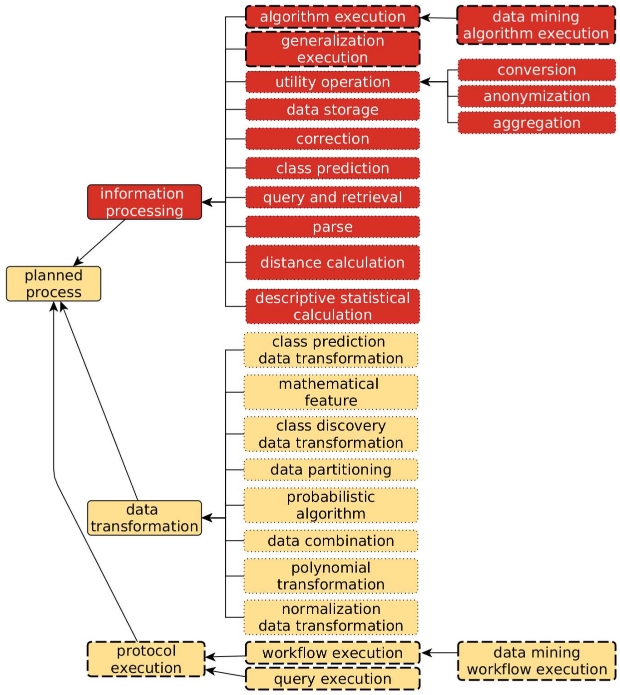

Figure 3.13 – OntoDM application level. Red boxes represent IAO classes. Yellow boxes represent OBI classes. Dash-dotted boxes represent OntoDM extension. Dotted boxes represent examples of class reuse from IAO and OBI.

entity between the information processing objective specification and the actual information processing execution. *Algorithm implementations* can play *roles* in certain KDDM processes. Therefore, OntoDM represents these *roles* and *data mining roles* by extending BFO *roles* entity. Lastly, the concept of *workflow* is defined as a specialization of IAO *plan* that concretizes (**isConcretizationOf**) *scenario*. *Workflows* have (**hasPart**) one or more *algorithm implementations*, representing realizable KDDM implementation. *Data mining workflows* are the concretization of *data mining scenario*.

APPLICATION LEVEL The application level of OntoDM-core consists of classes that are extensions of OBI class *planned process* and SWO class *information processing*, as illustrated in 3.13. OntoDM specializes SWO class *information processing* to represent the *algorithm execution*, which realizes an *operator*. *Data mining algorithm execution* specializes *algorithms execution*, which realizes a *data mining operator*, having *DM-datasets* as input and *generalizations* as outputs.

Generalization execution represents the application of a *data mining task* outcome, such as predictive models. Nonetheless, SWO defines a wide range of *information processing* classes that could be used to describe data processing algorithms in *virtual sensors*. We highlight some of these examples, using dotted box classes: *utility operation*, *data storage*, *correction*, *class prediction*, *query and retrieval*, *parse*, *distance calculation*, and *descriptive statistical calculation*.

Figure 3.14 presents OntoDM representation levels in five conceptual modules. In Figure 3.14.(a), a KDDM is represented using OntoDM-KDD

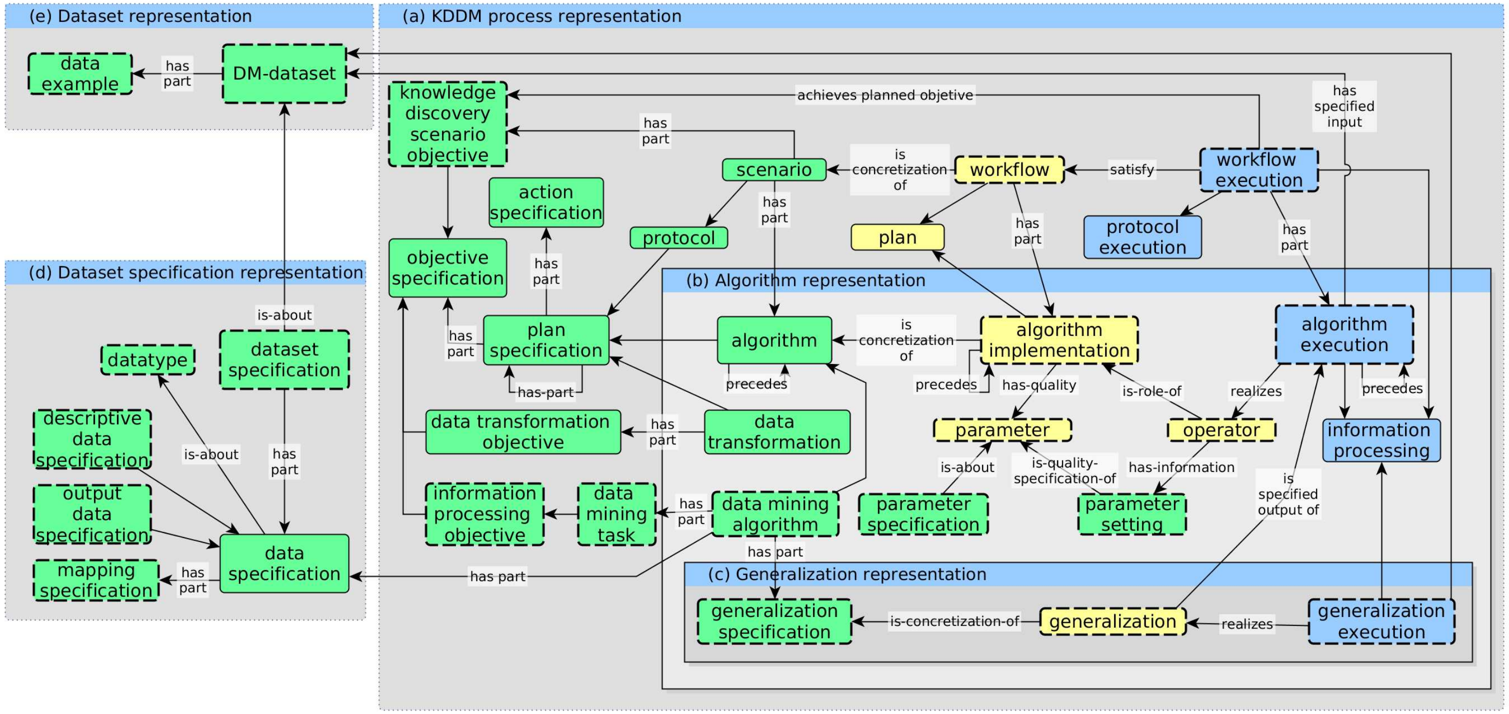

Figure 3.14 – OntoDM representation levels. Green boxes represent the specification level. Yellow boxes represent the implementation level. Blue boxes represent the application level. Dash-dotted boxes are OntoDM extended from reused classes of middle and upper-level ontologies.

classes. The *scenario* has (**hasPart**) a *knowledge discovery scenario objective* as *objective specification*, and, optionally, *action specifications*. The main composition of scenarios is the *plan specification* of the type *algorithms* and *data transformations*. A *workflow* is a OBI *plan* that concretizes (**isConcretizationOf**) a *scenario* and is mainly composed by (**hasPart**) *algorithm implementations*. Each algorithm implementation has a role (inverse property of **isRoleOf**) of some *operator*. Practically, algorithm implementations, such as those found in the WEKA toolkit, are implemented as *operators* by having *parameters* specified (**hasInformation** some *parameter setting*). Then, a *workflow execution*, which is a OBI *plan execution*, satisfies this *workflow* and achieves a *knowledge discovery scenario objective* through *algorithm executions* that realize each operator defined in that *workflow* (see Figure 3.14.(b)). *Algorithms*, *algorithm implementations*, and *algorithm executions*, following the same enchained concretization relation among *scenario*, *worflow*, and *workflow execution*. In addition, it is possible to represent precedence relation in the each of these levels of representation of algorithm entities.

OntoDM class *data mining algorithm* introduces a new level of abstraction in this recursive structure: the generalization representation. As explained previously, *data mining algorithms* produce *generalizations*, which can be conceived as a representation or a *generalization execution*, as depicted in Figure 3.14.(c). Each *data mining algorithm* has parts *data specification* that defines which part refers to *descriptive data specification* and *output data specification*, explicit by *mapping specifications*. In addition, *data specifications* are about *datatypes* and constitute *dataset specifications*, as illustrated in Figure 3.14.(d). Lastly, in Figure 3.14.(e), dataset representation consists of *DM-dataset* that are composed by *data examples*. *Algorithm executions* and *generalization executions* have specified *DM-dataset* as input and output, which are in turn specified (**isAbout**) according to *dataset specifications*.

3.4 BEHAVIORAL COMPUTING

As described previously, the term personal information is traditionally defined as all data, information, and knowledge related to an individual and/or under her control. The analysis, classification, and management of *personal information* is a challenging task due to its different aspects and perspectives. The Behavior Computing (BC) offers a systematic way of understanding features that explains or predicts behaviors, which can be exploited to represent personal information in contexts. BC aims to build formal methods and computational theories of behavior representation, processing, and engineering [21]. In BC, *behaviors* refer to actions, operations or events conducted by agents within certain context and environment (virtual or physical ones), focusing on symbolic behaviors that represent these activities into a computational model. Thereby, it can be stated that BC intends to enrich the process of behavior pattern analysis, by modeling behavior features and allowing in-depth analysis of behavior and its impacts.

The research field of behavior analysis has been conducted based on scrutiny of data using KDDM techniques to recognize and identify behaviors. For instance, McIlwraith and Yang [134] identify key research areas for data processing within body sensor networks, such as dimensionality reduction, feature extraction, feature selection, and inference in order to observe human behaviors. The behavior recognition, identification, and selection are conceived as part of the design of the experiments. Thus, patterns and data generalizations are associated with behaviors based on the background knowledge of the experiment designers. However, no further contextual information about the real behavior is encoded. BC provides methods and representation to address these shortcomings of the traditional behavioral analysis. It focuses on the subsequent analysis phase and interpretation of this information, particularly meeting the objectives of our *privacy by design* model that aims to classify sensor data based on the personal information that can be inferred from it.

BEHAVIORAL MODEL Cao [21, 135] proposes an empirical behavioral model, defining behavior(γ) as a vector of the following key aspects called *behavioral features*:

— *Subject (s)*: entity that issues the activity (agent);
— *Object (o)*: entity on which a behavior is imposed;
— *Place (w)*: location where a behavior happens (spatial aspect);
— *Context (e)*: environment in which a behavior happens (it may include pre-condition and post-condition);
— *Belief (b)*: the subject's informational state and knowledge by which the subject conceives the world in the moment of the observed behavior;
— *Action (a)*: actionable task being executed by the subject during the observed behavior;
— *Goal (g)*: objectives intended to be accomplished by the subject;
— *Plan (l)*: sequences of actions that the subject can perform to achieve one or more of its intentions;
— *Impact (f)*: the results achieved by the execution of a behavior on its object and/or context;
— *Constraint (c)*: conditions that impact on the behavior;
— *Status (u)*: stage on which the behavior is currently stated;
— *Associate (m)*: relationship of interaction or impact between distinct behaviors; and
— *Time (t)*: time when a behavior occurs (temporal aspect).

According to these behavior features, a behavioral vector ($\vec{\gamma}$) can be represented as follows:

$$\vec{\gamma} = \{s, o, w, e, b, a, g, l, f, c, u, m, t\} \tag{3.1}$$

These features are extensive to describe behaviors in different contexts and are not intended to be a minimal set of features. In fact, this vector of behavior features describe classes of all possible physical, relational, and abstract elements that one needs to describe any behavior regardless of the domain of interest. The BC framework has been used in several works [136, 137, 138, 135, 139]. In [140], the authors represent behaviors using causal relationships, representing the causality among behavioral features and demonstrating an improvement in activity prediction and user identification. Besides the spatio-temporal, social, organizational and environmental aspects related to the behavior ($\vec{\gamma}$), a behavior sequence can be represented as an ordered vector $\vec{\tau}$ of behaviors as follows:

$$\vec{\tau} = \vec{\gamma}_1, \vec{\gamma}_2, \ldots, \vec{\gamma}_n \tag{3.2}$$

This representation attempts to provide a richer representation for behavior analysis, capturing features that participate in the observed event, the relationship between these features, and the relation between a behavior and its subsequent one. Personal information is, therefore, intrinsically represented in this model, since it is produced, used or managed in the behavioral context. This sequence also provides a more comprehensive and informative vector-oriented for behavior pattern analysis. The BC includes nine research axes [21]:

— *Behavior data construction*: consists in transforming data into *behavioral data*. This transformation intends to associate behavioral entities to each data. Semantic annotation and mapping functions are studied in this axis;
— *Behavior modeling and representation*: corresponds to formal methods and techniques to capture behavioral entities, their attributes, and properties, as well as their relationship among themselves. Modeling languages, such as OWL, can be used to represent these entities and to reveal the interaction, causality, evolution, divergence, convergence of behaviors;
— *Behavior pattern analysis*: refers to the extraction of patterns based on the behavior modeling and representation. Classical KDDM techniques are exploited to extract information from behaviors through behavior clustering, cause-effect analysis, behavior streaming mining, activity mining, correlation, and so forth;
— *Behavior impact analysis*: refers to the analysis of relational causality based on the representation of behavior. Techniques to detect, predict, prevent behaviors are supported by impact modeling, impact analysis, risk analysis, and other tools to measure the behavior impact;
— *Behavior network*: consists in analysis of patterns, flocks, and rules based on the network and its dynamic nature of individual and crowd behavior;
— *Behavior simulation*: represents the investigation of behavior simulation aiming to anticipate consequences and scenarios to base decision making. Multiagent intelligent-based simulations are part of this research axis;
— *Behavior measurement and evaluation*: corresponds to techniques to quantify and evaluate the impact and behavior networks to measure the significance of behavior patterns. Specific measure-

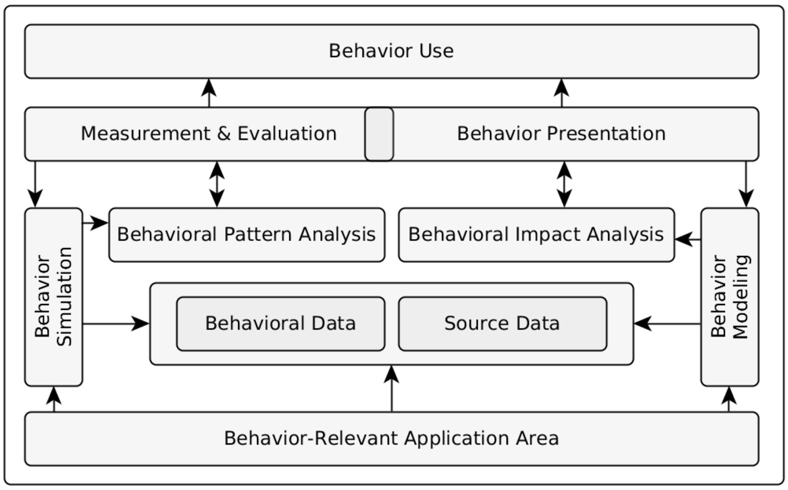

Figure 3.15 – Behavior Computing Research Axes [21]

ment such as utility, privacy harm risk, costs, and organizational image impact are some example;

— *Behavior presentation*: consists in techniques to present all the information up-mentioned. Similarly to data mining, challenges to present patterns and statistical models, behavior presentation aims to adapt this information in a visual presentation;

— *Behavior use*: represents the research axis that investigates how this information can be practically used. The possibility to represent and reason about behavior allows incorporating this intelligence into systems that will respond more accordingly to human behaviors.

Figure 3.15 further illustrate these research axes and the relation among them. The use of BC to represent personal information involves more than one of these axes. In particular, the *behavioral modeling and representation* and *behavior use* are aligned to our needs of representing and reasoning personal information using Semantic Web technologies.

Cao's model provides a flexible way to structure *personal information*, since its behavioral features, that participate in the observed behavior, can be related to the agent and extended according to the domain of usage. However, BC does not define which *behavioral features* should be used to describe behaviors. Consequently, behaviors can be represented according to any of the *behavioral entities*. For instance, if bike (*object*) is a central element for a mobility application, features that are important to the behavioral observation related to this object should be represented. Hence, it is indispensable to construct our definitions of *personal information* around the concept of *behavioral agent*. This premise ensures that once identified the behavioral temporal frame and the involved *behavioral agent(s)*, it is possible to classify the data as personal.

3.5 RELATED WORKS

Few attempts to describe *personal information* using Semantic Web and ontologies have been proposed. However, most of them do not cover the IoT sensing issues, such classification of sensor data and its contextual information. They target in the classification of general information artifact entities, such as files or documents. Some of these works are solely focused on the ontology definition, while other are proposed in the context of Privacy-Enhancing Technologies (PETs), which are covered in the next chapter. We refrain on describing two approaches for classification of personal information that are mainly based on ontologies.

In [141], a framework for representing personal information models – Personal Information Model Ontology (PIMO) – is proposed. The framework is defined using ontologies and focuses also on the conceptualization of personal information around the concept of an information owner. Personal information is defined in terms of projects, tasks, contacts, organizations, files, e-mails, and other resources of interest to the user to be categorized. The approach also employs upper-level and middle-level domain ontologies, aiming to provide a cognitive classification system for a personal knowledge workspace. The first constraint of this work consists of its exclusivity to personal workspaces, such as devices and computer applications. The approach does not address cases when the classification of information are challenged for end-user, such as the scenario of IoT and its sensor data. Another restriction comprises its inability to describe collective entities, such as a group of people or multiple owners. In addition, the class of personal information are defined by the end-users and are not associated with contexts, which hampers the possibility to automatize privacy policies based on user's context.

In [142], an ontological framework is defined using the concepts of points of views and organizing the reuse of domain-specific ontologies to classify and contextualize personal information. Each user has the possibility to design ontologies that will support the classification of personal information and contexts. The framework provides a formal definition for ontologies that represent: i) personal information, ii) information type and classification, and iii) personal and informational context. Based on these ontologies, a set of semantic mappings between different contexts and information type delivers a powerful inference that is used to enforce privacy. For this, the approach defines a web service composition that employs SPARQL queries to control access to personal information according to privacy policies and the set of ontologies defined by the user. Nonetheless, this ontological framework does not define axioms to classify personal information, restraining to specify formally the structure of the up-mentioned ontologies. Besides that, none of the concerns about privacy in IoT are addressed, such as classification of sensor data and detection of malicious inference intention.

3.6 CONCLUSION

In this chapter, we reviewed key concepts of Semantic Web that will be used to define our ontology for personal information and enabling technologies that will be used to define our *privacy by design* model. We discuss how Semantic Sensor Network Ontology (SSN-O) can be used as a starting point to understand the current issues in the classification of sensor data regarding privacy issues. We noticed its limitation to describe the information that flows in the IoT and the lack of information about data processing that is executed in the IoT platform during the sensing service.

Aiming to address these problems, we introduced the concepts of Meta-Mining (MM) and Behavior Computing (BC). For this, we introduced the concept of Knowledge Discovery and Data Mining (KDDM) and delimited the part that we intend to map in our ontology and privacy model. We investigated the most relevant ontologies to represent KDDM process and techniques, outlining the similarities and differences that guided us to select Ontology of Data Mining (OntoDM). We presented BC as the theory that will permit us describing personal information in behavioral contexts. The choice of this theory is justified based on its capacity to focus on the meaningful information that can be extracted from data. BC structures research axes and behavioral features that facilitate to model personal information. Lastly, we presented the most relevant works related to the definition of personal information using the Semantic Web technologies and ontologies.

4 PRIVACY ENHANCING TECHNOLOGIES AND APPROACHES

CONTENTS

INTRODUCTION

Several Privacy-Enhancing Technologies (PETs) attempt to guarantee privacy for data publishing. In the context of privacy in the IoT sensing, these approaches are incorporated intothe data stream, aiming to preserve data privacy *on-the-fly*. In this manuscript, we intend to propose a *privacy by design* model based on the principle of *plurality*, which assumes that several PETs can be opportunistically employed.

In this chapter, we investigate two categories of PETs: Privacy-Preserving Data Mining Technique (PPDMT) and Access Control Model (ACM). The former addresses the problem of privacy by degrading data utility of data stream and dataset. The latter enforces privacy by controlling access to private information based on the evaluation of privacy policies or rules. We intend to present an overview of these works and their key concepts to introduce the subject of privacy issues and strategies addressed by these technologies.

In Section 4.1, main categories of PPDMT are presented, along with works related to privacy preservation in data mining results and data streams. Next, in Section 4.2, we focus on relevant ACMs related to some enabling technologies that we intend to employ in our *privacy by design* model. Lastly, we conclude with an overview of these approaches and a brief analysis.

4.1 PRIVACY-PRESERVING DATA MINING TECHNIQUES

In the *recipient sphere*, published or streamed data may be exploited by privacy adversary to extract private information. While ACMs offer binary decisions for granting or denying authorization to perform operations on private information, Privacy-Preserving Data Mining Techniques (PPDMTs) may provide a more flexible solution to preserve privacy. PPDMT aims to preserve privacy in published or streamed data by degrading data utility selectively in a way that remains practically useful for KDDM scenarios.

The most basic form of PPDMT is based on the perturbation approach. Randomization or data perturbation is a technique for modifying data adding noise generated by a random process. Sensitive data can be perturbed by transforming it using functions, such as add, subtraction, nullification, or any other mathematical function. Traditionally used in the context of distorting data by probability distributions, randomization methods were applied to data mining techniques as an initial approach for anonymization. This perturbation configures a challenge for data mining techniques since they are inherently dependent on statistics. The main advantage of this method is that it is relatively simple. An example of randomization methods is the *multiplicative perturbations* that focus in reducing data dimensionality by perturbing multi-dimensionality. Its main flaws rely on adversarial attacks where the privacy adversary knows some linearly independent collection of the records or data samples. Another type of randomization is based on *data swapping* that aims to preserve privacy by swapping values of different records.

More sophisticated privacy-preserving techniques were proposed, so-called anonymization methods, based on the concept of hiding groups of anonymous attributes, making harder for privacy adversary attacks to distinguish between individuals and, consequently, the association between privacy information and the concerned individual. The anonymization problem can be generically defined based on the following types of attributes:

Sensitive content			
explicit identifiers	quasi-identifiers	sensitive attributes	non-sensitive attributes

Explicit identifiers are the set of attributes that contain explicit information about individual's identity, such as Social Security Number or name. *Quasi-identifier* is a set of attributes that could be correlated to explicit identifiers and potentially discloses individual's identity. *Sensitive attributes* refer to sensitive information other than identity, such as medical records, religion, and salary. *Non-sensitive attributes* consist of all other attributes that previously classified. Therefore, *anonymization* refers to approaches that seek to remove all explicit sensitive content and hide all statistical traces that could be correlated to *quasi-identifiers*.

A myriad of PPDMT has been proposed in the literature. In fact, the *anonymization-reidentification* cycle described in Section 2.2, is probably the main reason why so many variants of anonymization techniques have been proposed. Most of these techniques are implementations of same strategies to remove traces from the published data that would allow correlating it with private content.

Anonymization attacks can be classified generally into two groups [143]: *linkage-based* and *probabilistic-based*. *Linkage-based attacks* consist of threats that envision to link record, attributes or tuples to a sensitive content based on explicit identifiers or quasi-identifiers. These attack models are called *record linkage*, *attribute linkage*, and

Privacy model	Attack Model			
	Record linkage	Attribute linkage	Table linkage	Probabilistic attack
k-Anonymity	•			
MultiR k-Anonymity	•			
l-Diversity	•	•		
Confidence bounding		•		
(α,k)-Anonymity	•	•		
(X,Y)-Privacy	•	•		
(k,e)-Anonymity		•		
(ϵ,m)-Anonymity		•		
t-Closeness		•		•
δ-Presence			•	
(c,t)-Isolation	•			•
ϵ-Differential Privacy			•	•
(d,γ)-Privacy			•	•
Distributional Privacy			•	•

Table 4.1 – Privacy-Preserving Data Mining Technique (PPDMT) [143]

table linkage respectively. *Probabilistic attacks* refer to attacks that use statistical hypothesis tests to discover identifiers or quasi-identifiers based on the *uninformative principle* whose goals is to ensure that the difference between the prior and posterior beliefs is small. For the matter of brevity, we refrain on describing the main categories of anonymization techniques surveyed in [143], and summarized in Table in Table 4.1 according to the up-mentioned attack models. Further information about PPDMTs can be found in [144, 54, 145, 146, 147].

In the *record linkage* attack model, the privacy adversary tries to identify uniquely a record that belongs to the victim. Therefore, the k-anonymity approach consists in hiding this identification in $k-1$ records, i. e., in a k-anonymous table, each record is indistinguishable from at least $k-1$ other records with regard to the sensitive content. In (X,Y)-anonymity, X and Y are disjoint sets of attributes, where each element in X is linked to at least k elements in Y. k-anonymity is a single case in (X,Y)-anonymity where Y is a surrogate key in the table. *MultiR* k-anonymity extends the traditional k-anonymity to a multi-table scenario, where the uniqueness of the joint-tuple must be guaranteed.

In *attribute linkage* attack model, even if the adversary cannot precise the record of the victim, it will try to infer private attribute from the published or streamed data based on group association. Therefore, the diversity principle may prevent the attribute linkage. The l-diversity technique requires that every group that shares a specific quasi-identifier contains at least l other sensitive attributes. In order to puzzle the occurrence of the other sensitive attributes, minimizing the chances of identifying the l diversity group, the (c,l)-diversity guarantees a normal distribution to these values. Confidence bounding techniques assure the maximum level of confidence of inferring a sensitive attribute on a group of one or more quasi-identifiers, focusing on the confidence probability instead of group size. Similarly to the application of confidence bounding to k-anonymity, the (X,Y)-Privacy combines the confidence bounding concept to (X,Y)-anonymity, controlling the maximum level of confidence of inferring Y (and consequently X). The (α,k)-anonymity is similar to (X,Y)-Privacy approach, except by the fact that X are always quasi-identifier attributes.

Since most works on k-anonymity consider only categorical sensitive attributes, (k,e)-anonymity proposes to anonymize numerical values by dividing groups containing at least k different sensitive values with a range of at least e. Similarly, (ϵ,m)-anonymity proposes to guarantee the privacy of numerical sensitive attributes, but limiting the confidence of inferring values in a given range $[value - \epsilon, value + \epsilon]$. In order to prevent skewness attack, i. e., when the chances of privacy attack increase because of the overall asymmetric distribution of quasi-identifiers, *t-Closeness* approaches propose to normalize the distributions of sensitive attributes to the same t level.

In the *table linkage* attack model, the privacy adversary intends to identify the presence or absence of a record in the table. To prevent this attack, the δ-Presence techniques propose to bound the probability of inferring the existence of record within a specific range δ.

The *probabilistic attack* model focuses on systematically assuring a level of uncertainty about a prior belief of a privacy adversary. By believing that the background information does not influence in the privacy attack, (c,t)-Isolation approaches prevent *record linkage* guaranteeing a radial distance between the real private attribute value and its inferred counterpart value sought by the privacy adversary.

The ϵ-*Differential privacy* is currently called the silver bullet of the PPDMT and consists in guaranteeing that is safe for a user to share her information, guaranteeing that one single record containing sensitive content in the published dataset or data stream will not harm her privacy. These approaches aim at calculating the probability difference between the results obtained by executing a randomized function over the records of the table that contains the private record and the table that does not. The logarithm difference should not be greater than the ϵ threshold. On top of this reasoning, (d,γ)-Privacy approaches propose to bound this threshold to a data utility measurement, respecting a reasonable trade-off between privacy and utility. (d,γ)-Privacy approaches assume that for each record in the published dataset, the probability of privacy adversary identifying this record is not greater than d. At last, *distributional privacy* increments the model of ϵ-*Differential privacy* aiming to make sure that the published dataset reveals only information about a specific underlying distribution.

Another important strategy in privacy preservation takes into account the data decentralization to minimize the chances of a compromised data source or a data center exposing sensitive information. A frequent scenario involves groups of stakeholders who may be interested in performing data mining on the union of their datasets, but are restricted to share their data for legal, commercial or privacy policy reasons. The strategy aims at partitioning the data vertically or horizontally. In the vertical partitioning, each party a has different subset of attributes describing a common instance. In the horizontal partitioning, each party has the same set of attributes, but a subset of distinct instances. The main idea of these approaches is to encrypt data before sharing so the dedicated data mining algorithm can decrypt and works with the encrypted data in order to provide results to all the involved parties. Several approaches propose data mining techniques in partitioning strategy to preserve privacy, such as found Bayesian Network [148], clustering [149], support vector machines [150], decision trees [151], and so forth. The major advantage of the partitioning strategy relies on the lossless data quality, since the dedicated data mining algorithm can decrypt and use the original data set, only exposing its results. Its drawback remains the computational overhead added to encrypt, decrypt, and aggregate separate datasets to provide data mining [152]. In addition, the inability to classify and control access to data mining result may expose unintended information.

4.1.1 Privacy-Preserving Data Mining for Data Streams

Several approaches incorporate the up-mentioned PPDMTs works in the data stream to provide anonymization to degrade the data utility of KDDM results. A comprehensive survey of privacy model for big data is presented in [153]. Victor et al. [153] identify anonymization approaches for social network data, stream data, and differential privacy using Map-Reduce distributed strategy for data processing [154] and anonymization [155]. along with the anonymization approach, privacy model and addressed privacy issues. Data stream anonymization is particularly addressed by the Continuously Anonymizing STreaming data via adaptive cLustEring (CASTLE) approach [156] and Sensitive Attribute Bucketization and REdistribution framework (SABRE) t-closeness [157]. These approaches intend to calculate privacy measures and data utility factors for data mining generalizations in order to apply the same strategies exploited by traditional PPDMTs. However, by restricting the search space of private information through generalization, these approaches may increase the probability of privacy breaches. Bhattacharya et al.[158] study this limitation preliminarily but further results are needed to confirm that degrading data utility in data mining generalization are as efficient as traditional PPDMT approaches.

That et al. [159] propose PAMPAS, a privacy-aware mobile participatory sensing system for efficient mobile distributed query processing that collects, aggregates, and extracts information from geographic location data. PAMPAS implements a two layer architecture that enhance the security at the device level with its secure probe while orchestrating the data distribution and partitioning in a centralized supporting server infrastructure. The secure probes collect and encrypt data at the first moment, and can also be selected to decrypt and process the sensor data collected by other secure probes during the partitioning phase. The communication between probes and central supporting servers relies on the anonymized network, similarly as found in the Tor network, hiding user's identity in the crowd, as a premise to provide its privacy-aware sensing service.

4.2 ACCESS CONTROL MECHANISMS

Access Control Models (ACMs) consist of authorization mechanisms that allow specifying policy conditions to grant or deny authorization to execute an operation over resources. Basically, these models have four main elements: policy conditions, operations, resources, and the requester. Conditions represent the circumstances when the operation is authorized to a specific requester over a specific resource. Operations are commonly described in terms of access, modification, creation, and so forth; normally associated with the type of resources, such as systems, devices, and information. Requesters represent those users who demand access to resources. These users can be represented by individuals, groups, or roles profiles. ACMs offers contestability by design. The formal evaluation of conditions constitutes an unambiguous way to decide about granting or denial of resources and operations *on-the-fly*, but also posteriorly, in a privacy audit for instance.

There exist several types of ACM available covering different aspects of these four elements. For instance, in Mandatory Access Controls (MACs), an access policy is controlled by a security policy administrator, on which users do not have the ability to delegate or grant access to their own resources. Conversely, Discretionary Access Controls (DACs), commonly found in modern operating systems, allow users to modify or override permissions, such as access to files and directories. Another well-adopted ACM is the Role-based Access Control (RBAC) [160] which is defined around the concepts of roles and privileges, relying on the simplicity to assign permissions to roles, rather than for each user. Similarly, the Attribute-based Access Controls (ABACs) [161], an access control paradigm, offers a fine-grained policy condition definition, addressing some of the shortcomings that RBAC can generate regarding its unmanageable set of roles. In fact, the attribute granularity of ABAC allows defining policy conditions based on attributes of resources, requesters, and operations in a logic expression [162], instead of roles, which facilitate to express policy rules and compress the number of conditions. A survey of main ACMs can be found in [163].

Chronologically, first contributions for privacy in ubiquitous systems were proposed in [164] based on anonymous and secure connections to data source devices. This approach intended to inform *data providers* about who had or was requesting access to protected devices, evaluating privacy policies that are contextualized only by location and proximity. Similarly, in [165] a role-based privacy-aware access control is proposed for context-aware applications on mobile agents. The concept of policy rules is analyzed based on the context where the application requests personal information expressed in pairs of key values, such as location/office, social role/friend. A set of conditions defines a privacy rule to permit or deny access to private resources. These key values represent static conditions that are evaluated without any further knowledge about its content.

Ardagna et al. [166] propose an approach for privacy-aware access control based on a framework provides an expressive policy definition. The framework and the policy conditions encompass not just the access request evaluation but the subsequent usage of this information. The concept of certified and uncertified is also employed in this approach, classifying the platform to which data is going to be released. Each rule identifies subject (requester), object (information), purpose, conditions, and actions. The subject can be a specific request identifier or an expression to define the requester specifications based on a data structure, such as `requester.age`, `requester.nationality`, and so forth. The object can be a specific attribute or an expression that specifies a set of information, such as `object.createdAt > 2015`. Table 4.2 presents some example of access control policy rules and data handling policy rules and their descriptions.

The purpose describes the intention of how data are going to be used. The conditions correspond to run-time conditionals to be evaluated, such as a non-empty form, trusted platform, or any boolean function. The concepts of *data handling* and data handling policies allow specify the strategy in terms of processing sphere (server-side, user-side, or customized), access level (identity-based, category-based, attribute-based), meta-data type, restrictions based on provision (pre-conditions) and obligations (during and pos-conditions). However, the representation of data processing and KDDM are defined hard-coded, and thus, limited to code templates that must be respected to be compatible with its safe environment. Considering how dynamic IoT scenarios can be, the idea of providing a set of data mining templates or, more generically, pre-defined KDDM processes would limit the development of IoT applications. Additionally, this type of solution extends to certified platforms, which are trusted to produce personal information and to permit the custodian by data consumers following a specific privacy policy, is not suitable for the IoT sensing scenario of release-and-forget where data will inevitably be manipulated and exploited according to data consumer's need.

Along with the advantages of the Cloud Computing and its successful XaaS model, some privacy as a service model were proposed. In [167], Service Level Agreement (SLA) are proposed to specify trust model and data privacy. The concept of software execution and the data processing protocol are similar to the previous approach, limiting trusted KDDM applications to extract information using private datasets, which are encrypted in trusted Cloud platforms. The strategy consists in releasing only a subset of private datasets, decrypting according to privacy categories defined in the SLA, and allowing only authorized *data consumers* to have access to the output. *Data providers*

Rule	Description	
Access Control Rule #1	any WITH credential(employeeCard(equal(**user**.job,'Director')),ACME) AND declaration(equal(**user**.company, 'ACME')) CAN **read** ON cc_info WITH greaterThan(**object**.expiration,*today*) FOR {marketing,service_release} IF {in_area(**user**.sim,'ACME') and log_access()}	ACME's directors are authorized to read valid (i. e., not yet expired) cc_info for marketing and service release purposes, if they are located inside the ACME building and the access is logged.
Access Control Rule #2	any WITH credential(employeeCard(equal(**user**.job,'Seller'), equal(user.jobLevel,'A')), ACME) AND declaration(equal(**user**.company, 'ACME')) CAN **read** ON cc_info WITH greaterThan(**object**.expiration, *today*) FOR service_release IF log_access()	Sellers of level A of ACME are authorized to read valid cc_info for service release purpose if the access is logged.
Access Control Rule #3	any WITH credential(employeeCard(equal(**user**.job, 'BusinessConsultant')), ACME) CAN **read** ON cc_info WITH greaterThan(**object**.expiration, *today*) FOR reimbursement	ACME's business consultants are authorized to read valid cc_info for reimbursement purpose. Table
Data Handling Policy Rule #1	declaration(EQUAL (**user**.type, 'BusinessPartners')) CAN **read** FOR market PROVIDED pay_a_fee()	Business partners of ACME can read for market purpose the name and the contact info of user provided that they have paid a fee.
Data Handling Policy Rule #2	declaration(EQUAL(**user**.type, 'BusinessPartners')) CAN {read,write} FOR service_release FOLLOW delete_after(30 days)	Business partners of ACME can read and write for service release the name and the contact info of Alice. The name and contact info must then be deleted after thirty days.
Data Handling Policy Rule #3	declaration(EQUAL(**user**.type, 'MarketAgencies') AND **credential**(IMB_Cert(equal(**user**.speciality.category, 'computer')), IMB) CAN read FOR statistic	Market agencies specialized for distribution of computers and whose specialization has been certified by the International Market Board (IMB) authority can read the name and the contact info of the user for statistic purpose.

Table 4.2 – Examples of access control and data handling policy rules [166]

have only SLA to specify privacy policies to their sensitive data, being completely agnostic about the KDDM process executed by trusted softwares and their results. In addition, no further assumption about the context and contestability about the KDDM process execution can be inferred.

In [168], a *privacy by policy* approach was developed using ontology and Semantic Web technology. The approach defines a privacy by policy enforcement based on privacy protection ontologies and domain ontologies, following privacy engineering principles. The solution is based on queries that retrieve PEPs according to conditions as shown below:

```
Context(node-type='server' and requestor='TrafficStateApp')
{
        Permit process-query On location, trafficstate, vehicle-
            type As query-result
        Permit retrieve On query-result With (k-anonymity > 10)
        Permit TableAnonymization(metric='k-anonymity', anonymity
            -value='10') On query-result Retrieve=true
}
```

The approach proposes also a mechanism to define and evaluate privacy risks that are used to detect privacy issues in order to select, configure, and execute PETs. These privacy indicators are calculated based on the application properties and privacy engineering principles that describe aspects of the performed data processing and the application design, such as personal information and the number of performed functionalities, privacy principles, performed operations on personal information. In order to minimize the ambiguity of these concepts and calculate these indicators formally, an ontological framework is proposed constituted by three ontologies: *ICT base ontology*, *ICT privacy ontology*, and *ICT privacy protection ontology*. The *ICT base ontology* specifies concepts of information, data, system, and other concepts related to ICT and the application domain. The *ICT policy base ontology* contains the description of policy elements, such as privacy policy, policy statement, context, resource, permission, and condition. The *ICT privacy ontology* extends concepts from these both ontologies to represent data controllers, data processors, personal information, and so forth. The *ICT privacy protection ontology* provides a knowledge model for PETs, its inputs, outputs, concepts and parameters.

This ontological framework provides a domain-level ontology that needs to be extended to describe private information and PET. By associating *ICT base ontology* class individuals to *privacy ontology* and *privacy protection ontology*, the approach is capable of reasoning which PET should be performed depending on conditions based on domain specific concepts and applications data processing. Although the approach does not provide insights of context based on IoT sensing, such as behavior or personal workspace, it provides a base that could be exploited to specify such concepts. An example of an instance of this framework is presented in Figure 4.1. A data processing model is depicted in Figure 4.1.(a) composed by system, component, operation, access, process, result item (operation output),

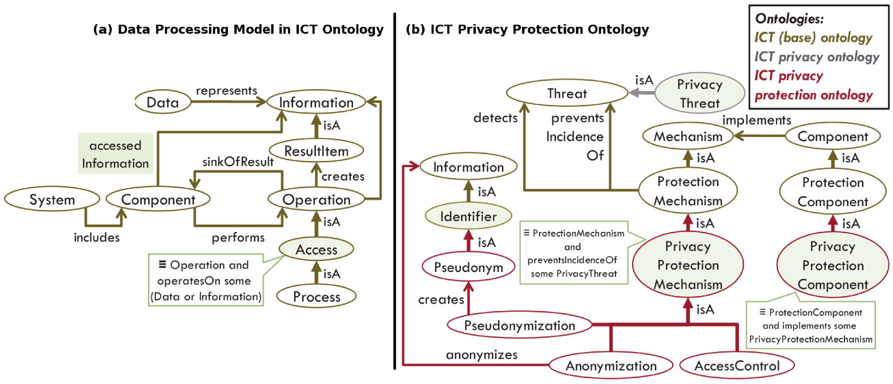

Figure 4.1 – Ontological framework for privacy preservation in location-based systems. [168]

data, and information. In Figure 4.1.(b), an instance of privacy protection ontology that associates information to PETs (privacy protection mechanisms) through concepts of PPDMT (anonymization and pseudo-anonymization). These PETs prevent or detect threats and are implemented by privacy protection components.

A context-aware privacy for sensor data on mobile systems called ipShield is proposed in [169]. The approach consists of controlling the application access to sensor data based on activities of these applications, using a similar strategy used in anti-virus applications. The ACM is implemented using the concept of a *semantic firewall* that detects patterns in the application activity and applies privacy rules according to user's privacy policy. Figure 4.2 presents the ipShield data flow. The sensor data (A) stream is monitored aiming to alert for suspicious occurrences of sensor usage (F) and detection of labeled contexts using machine learning techniques (D). IpShield provides two configuration mode: fine-grained and white/black list. The former analyzes the sensor data streaming activity to detect inferences and present for each inference, its label, number of occurrence and application. The latter create a privacy policy (G) reasoning over an inference knowledge base (E) and the sensor data stream activity. The user has the option to choose the configuration mode (J) and, consequently, the set of PETs that will be performed (B) before releasing sensor data to the application (C). In general, the privacy rule is expressed as condition (antecedent) and action (consequent). For instance, `((TimeOfDay in [10am : 5pm]) AND (Place = school) AND (AppName = facebook))` then apply `Action=Suppress on SensorType=gps`. The approach offers a limited set of actions, such as perturbation, suppression, constant value, and a strict of contexts, such as time of day, day of week, place, and external contexts provided by the machine learning techniques. The idea of using machine learning techniques to detect malicious application activities in the mobile platform benefits from the KDDM domain to enforce privacy. However, as the number of computation of inference labels *on-the-fly* increases, so does the usage of device resources. Moreover, the approach is application-centric and has

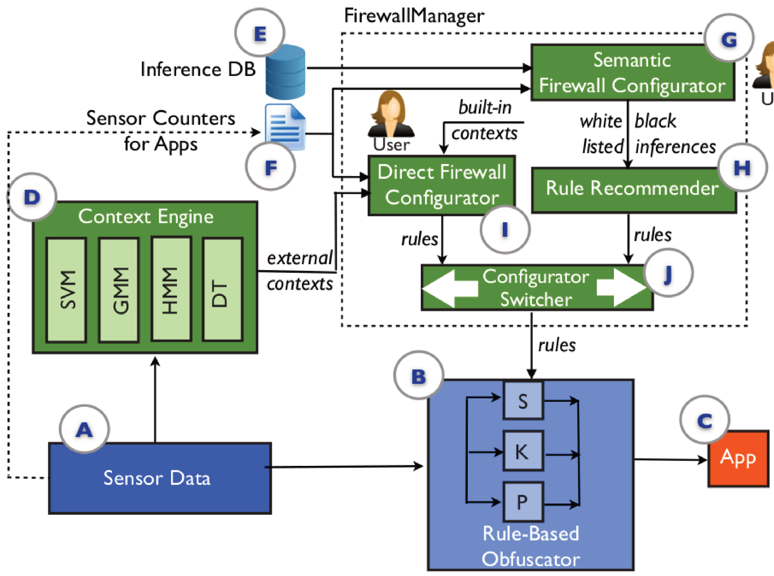

Figure 4.2 – ipShield dataflow [169]

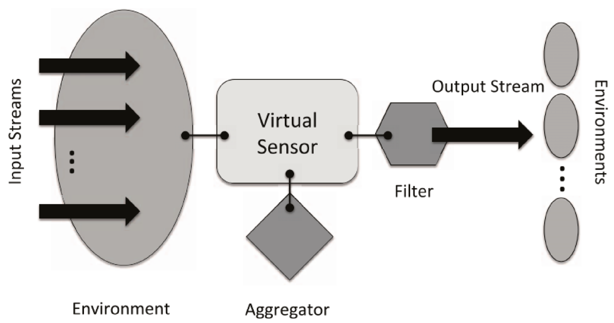

Figure 4.3 – Privacy Preserving Virtual Sensor Model for Social Mining [170]

a tardy protection, which is adapted to the problem of privacy on mobile platforms, but not ideal to the IoT sensing scenarios.

A privacy preserving model based on virtual sensor for ubiquitous social mining applications is proposed in [170], providing a modular *privacy by design* solution. The virtual sensor is implemented along a post-execution filter, controlling access to sensitive personal information that is outputted, and, thus, implementing a PEP for each sensor that streams through it. The virtual sensor model is presented in Figure 4.3, composed of its environment, an aggregator, a filter, and output stream. Similar to the GSN concept, this virtual sensor can have multiple data streams and data processing capability (aggregator) to process data *on-the-fly*. The *privacy by design* mechanism consists in enforcing privacy for each virtual sensor output, implementing a modular self-determination PEP in the IoT sensing. The approach is implemented in the mobile platform in the context of an IoT scenario to support crowd-sourced participation, exposing a generic interface defined by a set of functions, parameter, and information for social mining applications. Nonetheless, this privacy design does not foresee malicious virtual sensors that can gain access to sensor

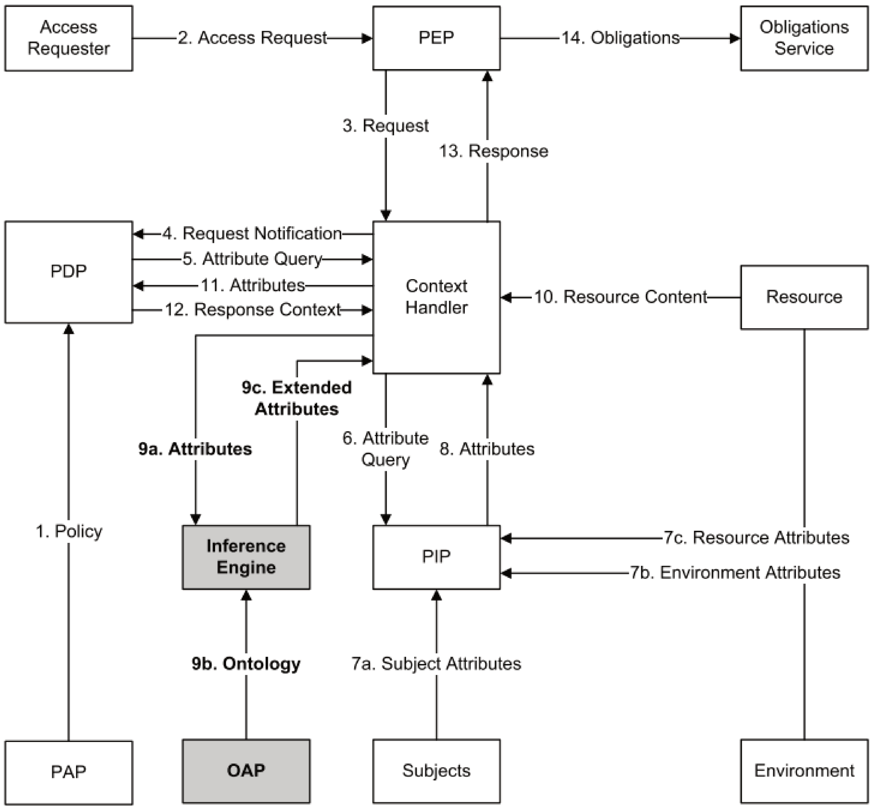

Figure 4.4 – Extended XACML architecture using Semantic Web technology [171]

data before its privacy enforcement. This lack of preventive detection of malicious inference impacts the IoT sensing performance and efficiency and opens security breaches for privacy adversaries to exploit ways to bypass the PEP.

The eXtensive Access Control Markup Language (XACML) standard is an important PEP that proposes to define a common vocabulary for ACMs and a standard architecture. For the matters of brevity, we focus on mentioning some relevant works from the extensive literature developed around it in order to describe its concepts, workflow, and shortcomings. In [171], an extension of the XACML architecture is implemented by employing Semantic Web technology to interpret resource contents. The extended architecture is presented in Figure 4.4. In the first step, the Policy Administration Point (PAP) generates an XACML policy and makes it available to the Policy Decision Point (PDP). At this state, the Privacy Enforcement Point (PEP) is ready to receive access requests (2) and forward them to the Context Handler (3). After transforming the request into an XACML request (4), the Context Handler sends it to the PDP. In the original XACML architecture, if the PDP needs more information to evaluate the request (5), the Context Handler would retrieve it from the Policy Information Point (PIP) (6-7), optionally retrieve the resource value (10) and would send everything directly back to the PDP (11). The approach in [171] provides the capacity to express the subject matters as a semantic representation and additional steps (9.a, 9b, 9.c) to be performed in order to retrieve all inferred values of such semantic representation, which can be class subsumption, SWRL inference or SPARQL query. Next, the PDP evaluates the policy and notifies the Context Handler of its decision (12), which, in turn, would translate it back to the PEP (13). Then, the PEP would verify if the request satisfies all obligations (14) before responding to the requester. This evaluation response, however, it is restricted to grant or deny access to resources.

In [172], an XACML-based approach is proposed for access control on the *data provider*'s side. The XACML semantics are extended to represent the service type and contextual elements of the private resource, such as purpose, recipient, retention, and purpose of usage. The extended representation improves the XACML expressiveness of both privacy policy conditions and classification of private information. The approach also provides predefined intention of data usage: read, collect, and share. A first step towards the creation of a richer semantics for data operations and a classification for private information is proposed, although as a predefined structure.

4.3 CONCLUSION

In this chapter, we presented Privacy-Enhancing Technologies (PETs) and works related to our proposed *privacy by design* model. We introduced the main categories of Privacy-Preserving Data Mining Techniques (PPDMTs) and a brief presentation of works that implement these privacy-preserving strategies to KDDM results. From a privacy engineering perspective, these approaches do not provide *contestability* due to its premise that data is published in an untrusted environment where privacy adversaries can exploit underlying statistical information. However, this strategy permits to relax the trade-off between privacy and data utility, which is binary and strict in ACMs. We identify two main issues in PPDMT concerning the *privacy by design* model that we intend to propose. Firstly, the number of private attributed defined by the end-user can be numerous and may impact the performance of these PPDMTs on the sensor data stream. Secondly, the *untrusted* model that allows publishing anonymized datasets or data stream is one entry door for privacy attacks. We intend to address these two shortcomings of PPDMTs in or *privacy by design* model.

We also introduce key concepts of Access Control Models (ACMs) and related works that allow us to understand the main challenges of *privacy by policy* approaches and the most relevant works. Since our *privacy by design* model implements a *privacy by policy* mechanism, the review of such related works supported our process of definition of our *privacy by policy* mechanism. The ACM related works were analyzed from a perspective of privacy engineering and privacy preservation directives discussed in Section 2.2.

Part II

CONTRIBUTION

5 OPIS: AN ONTOLOGY FOR PERSONAL INFORMATION ON THE SENSOR WEB

CONTENTS

INTRODUCTION

In Chapter 2, the Semantic Web was identified as a key IoT enabling technology that can be used to leverage privacy enforcement in the IoT sensing. In this thesis, we address the problem of privacy enforcement in the IoT by providing an ontology-based *privacy-by-policy* mechanism sensing that anticipating data processing and evaluating privacy conditions using semantic annotation of sensor data, data analytics algorithms, and *personal information*.

In this chapter, we present our first contribution related to the development of OPIS, an ontology to represent *personal information* on the Sensor Web using concepts from the Behavior Computing (BC) and Meta-Mining (MM). For this reason, in Chapter 3, we introduced three knowledge representations: the Semantic Sensor Network Ontology (SSN-O), the Ontology of Data Mining (OntoDM), and the Behavior Computing (BC). The SSN-O constitutes the base on which we extend the concepts of sensor and feature of interest. The OntoDM is used to represent KDDMs processes, extending the concept of physical sensors into *virtual sensors*. The BC is used to model and represent *personal information* as behavioral features.

In the remainder of this chapter, Section 5.1 presents the rationale for designing the proposed ontology. Next, in Section 5.2, the goal, scope, and competencies of this ontology are described. Then, the ontology design choices, methodology, engineering principles, and guidelines used to develop the ontology are presented in Section 5.3. An overview of OPIS is presented, along with its conceptual layers and core concepts in Section 5.4. Next, knowledge models for personal information and behavior recognition are presented in Section 5.5, basing our behavior-centric model for personal information. In Section 5.6 the concept of information abstraction and SP are explained and used to define the concept of *virtual sensors* and to extend the concept of SP. The Personal Information Layer is presented in Section 5.7, followed by the Semantic Perception Layer (Section 5.8). We present examples, a use case, and the semantic queries in Section 5.9 to demonstrate the ontology competencies; and summarizes the contributions of our work in Section 5.10.

5.1 THE RATIONALE

The Semantic Sensor Network Ontology (SSN-O) [40] has become a *de facto* standard for semantic sensor annotation on the Sensor Web. It provides a semantic abstraction to describe sensor, observation, and features of interest, creating an interoperability layer and a common vocabulary to abstract these concept and deal with sensor data seamlessly. However, its semantic representation is not competent to describe the association of sensor data to inferred information and data features that arise in the context of information processing, such as level of confidence, the degree of freedom, accuracy, precision etc.

The intensive usage of data mining techniques to infer meaningful information from sensor data has partially introduced these issues. Traditionally, these techniques were executed over sensor data samples. More recently, with the dissemination of sensor data streaming, and the challenges created by its big data aspect, approaches for reasoning over data streams have become necessary if one intends to extract information continuously in a timely fashion [84]. This shifts the sensing process toward an *in-network* data processing paradigm, which can be defined as a reasoning layer between physical sensors and final applications. The virtualization of physical sensors addresses resource limitation issues, common in portable devices, such as storage and computing capacities, in the same time that encapsulates the complexity of information discovery and data mining implementations.

In fact, the manipulation and usage of high-level information, instead of raw sensor data, have been the main concern of the *perceptual computing* and the *cognitive computing* paradigms that are claimed to be the next stage of context-aware applications [69, 173]. In this context, Henson *et al.* [68] define the *perception computing* using the concept of Semantic Perception (SP) that aims to infer information by emulating the human perception. The SP generalizes observations in order to identify known patterns and abstractions that can be associated with a high-level information (perceptions) using abductive inference. For this, a background knowledge is expressed and associated with patterns, so these perceptions can be systemically derived.

Personal information is intrinsically related to the concept of information processing and human behaviors due to its extensive use of machine learning and data mining techniques to recognize human activities, behaviors and related information from sensor data stream [174]. The myriad of contexts and situations, on which this information is collected and interpreted on the Sensor Web, is too vast and has not been covered by any Semantic Sensor Network ontology. In the case of the SSN-O, the concept of *feature of interest* represents an entity whose qualities are observed and associated with the context of an observation. No further representation for personal information or human context exists associated with features of interest as a consequence of the SSN-O design for domain agnosticism, pushing this definition toward applications and sub-domains. In addition, the ontology design privileges the sensing process context instead of the information discovery that produces or infers meaningful information from the sensor observation. In particular, the SSO pattern [121], proposed in the SSN-O, is not designed to express information process scenarios. SSN-O sensors can only represent systems that sense a stimulus to produce an observation about some feature of interest.

OPIS, an Ontology for Personal Information on the Sensor web is introduced in this chapter aiming to address these limitations of the SSN-O. For this purpose, the SSN-O concept of sensor (ssn:sensor) is extended to represent reasoning over sensor data stream as *virtual sensors*. Concepts from the OntoDM [131], a Meta-Mining (MM) ontology, are imported to represent a *virtual sensor* in terms of its process, implementation, and execution. In lieu of the SSO pattern, *virtual sensors* are modeled around the concept of SP. In order to represent personal information, features of interest are modeled according to the context of *human behavior*. In order to make this conceptual model suitable for a range of use case scenarios, we assert that personal information is collected, produced, managed, and controlled based on the context of behavior and its temporal frame. Therefore, the conceptual framework of the BC [21] is used to define the *behavioral entities*, a set of semantic representation for features of interests involved during a human behavior, that is capable of serving as a conceptual framework for personal information.

5.2 GOAL, SCOPE, AND COMPETENCIES

OPIS is a modular ontology for the Semantic Sensor Network. Its main goal is to provide an ontological framework capable of representing personal information on the Sensor Web, considering the behavioral context where this information is collected, managed, and used; and the information discovery scenarios that are specified, implemented and executed to produce it.

Q_n	Question
1	Which type of *personal information* (behavioral entity) is related to a sensor data **X**?
2	Which *virtual sensors* produce *personal information* **X**?
3	What is the *behavior* (context) related to the observed/perceived *personal information* **X**?
4	Which parameters are related to the quality of personal information using the *semantic perception process* **X**?
5	Which information processing techniques are used to infer *personal information* **X**?
6	Which information processing techniques are used in *virtual sensor* **X**?
7	On which (virtual or physical) machine a *personal information* **X** has been inferred?
8	Which properties of features of interest are used as input of *semantic perception process* **X**?

Table 5.1 – Competency questions

In order to describe *virtual sensors* and *personal information*, the concepts of sensor and feature of interest from the SSN-O are extended. *Virtual sensors* can be expressed in terms of specification (process), implementation, and execution. The process extends the SSN-O sensing process (ssn:sensingProcess), allowing to specify scenarios, algorithms, objectives, inputs, outputs, and data types. The implementation permits to describe how the process is concretized, along with run-time parameter specifications and values, toolkits, and programs. The execution represents how the implementation is concretely realized (executed) in a machine, as well as its input and output. The *personal information* is defined through *behavioral entities* that extend the SSN-O concept of feature of interest (ssn:featureOfInterest), its restrictions, and it is associated with *virtual sensor* inputs and output through the *Semantic Perception* paradigm. Therefore, the scope of OPIS is defined around the concepts of *semantic perception*, *virtual sensor* and *behavioral entities* that are described in detail in Section 5.6, 5.8, and 5.7 respectively.

The competency of an ontology can be evaluated by querying individuals based on the semantics provided. For this reason, a list of natural language questions is useful for defining clearly what wants to be represented using the ontology. Considering the goals and scope presented in this Section, we established a list of competency questions, presented in Table 5.1, which OPIS is designed to answer.

5.3 ONTOLOGY DESIGN

The development of a new ontology is an exhaustive process that aims to build a formal representation consensus over a knowledge domain. This demands rigor to ensure characteristics considered relevant to ontologies, such as vocabulary, robust structure, and foundational ontology alignment that can maximize the chances of its wide adoption.

We followed principles and best practices for ontology development from the OBO Foundry[1] and the NeOn Project[2]. The common principle in both projects is the reuse of existing ontologies that intends to unify the vocabulary. This principle was based on the perspective of standardization and developed to improve ontology design progressively while aligning reciprocally ontological frameworks around shared and reused vocabularies [175]. The use of foundational ontologies, middle-level, and domain-level ontologies to acquire established knowledge representation in the researched domain is strongly suggested in HCOME, a methodology for ontology engineering proposed by Kotis and Vouros [176]. Other OBO Foundry principles followed in the development of OPIS include to: i) define a clearly bounded subject-matter, ii) make use of coherent natural language definitions of top-level terms, incorporating cross-product links to other OBO Foundry ontologies, and iii) represent common relations that are unambiguously defined.

The scope and content of the ontology must coverage a specific subject domain, providing a balanced coverage on the subject-matter. For this reason, the reuse of middle-level and domain ontologies can provide an ontological framework to verify this balance. Moreover, well-adopted ontologies can offer a consensus over a high-level description of a domain of knowledge and support interoperability with other ontologies.

In that direction, the ontology proposed in this thesis adopts the SSN-O as the domain-level ontology and extends its capacity to represent *personal information*. As described in Chapter 3, SSN-O is aligned to the foundational ontology DOLCE-DnS UltraLite (DUL), which is a simplification of the upper-level ontology DOLCE [177]. DOLCE was conceived as a foundational ontology in the context of project WonderWeb, which settles DOLCE formalization, formal mapping to BFO, and practical use cases based on the description of web services, the application server for the semantic web, and web service infrastructure. In order to propose domain-level instantiation of DOLCE, DOLCE-Lite was proposed, aligned to DOLCE and extended using the Description & Situations Ontology (DnS) [178], which is designed to represent situations, contexts, hypothetical assumptions, and methodology applied to several domains. This extension with DnS is used as a plug-in to DOLCE and provides a cognitive ontology that can be used to describe situations and contexts, called DOLCE-DnS UltraLite: DUL.

By extending its classes to represent semantic sensor annotation, the SSN-O benefits of DUL commonsense language to represent features of interest. As a consequence, these features can be represented using the same cognitive commitment commonly found in information discovery scenarios. Ultimately, the inference of information, such as events, physical objects, agents, places, mobile objects, docu-

1. http://www.obofoundry.org/ (accessed on 26/04/2017)
2. http://www.neon-project.org/ (accessed on 26/04/2017)

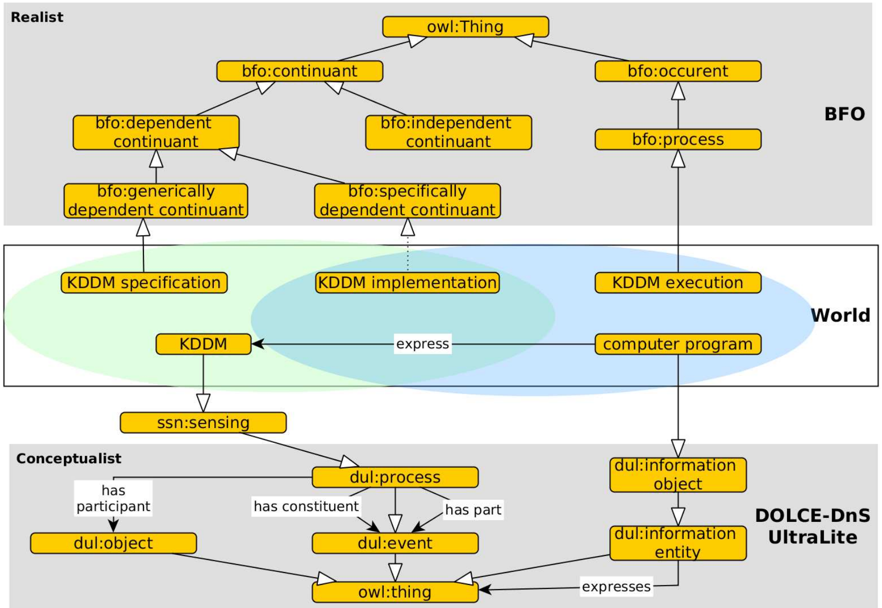

Figure 5.1 – An example to evidence the differences between DOLCE and BFO design choices.

ments, sentiments, is the objective of perceptual computing and cognitive computing paradigms used in context-aware applications that consume sensor data. On the other hand, the description of KDDM itself, its implementation and execution, cannot be specified using DUL. In this case, BFO-based ontologies such as the OntoDM offers a better representation, being capable of describing these entities on three important endurances: specification, implementation, and execution. This difference occurs because DOLCE (and DUL) is methodologically fundamentally conceptualist while BFO is essentially realist [179]. For example, DOLCE distinguishes between abstract and concrete entities, including agents and intention, being an *ontology of instances* that represent classes for these particular instances [112]. In contrast, BFO is committed to representing classes for particulars and for universal concepts, which allows representing temporal or spatial models while in DOLCE, these temporal and spatial models are not built natively. To cope with this discrepancy between the design commitments of SSN-O and OntoDM foundational ontologies, we propose to extend classes from both domain ontologies simultaneously.

In Figure 5.1, we present an example that evidences the difference between DOLCE and BFO in the context of our ontology design. The entities KDDM and its representation in different time endurance are not captured by DOLCE (SSN-O and DUL). It concentrates in describing KDDM as a process and a computer program (subclass of *information object*) without differences of endurance between its specification, implementation, and so forth. From a BFO realistic perspective, KDDM as a specification of a process can be expressed in two level of specification: KDDM specification (subclass of *generically dependent continuant*) and KDD implementation (subclass of *specifically dependent continuant*). The capacity to express a specific dependence to any *dependent con-*

Level	Origin	Namespace
Upper-level	DOLCE-DnS UltraLite	**dul**
	Basic Formal Ontology [3] version 1.1	**bfo11**
	Basic Formal Ontology version 2	**bfo2**
	OBO Relational Ontology [4]	**ro**
Middle-level	Ontology for Biomedical Investigation [5]	**obi**
	Information Artifact Ontology [6]	**iao**
	Software Ontology	**swo**
Domain	Semantic Sensor Network Ontology	**ssn**
	Ontology of Data Mining (OntoDM)	**odm**
	Environment Ontology [180]	**envo**
	Ontology for General Medical Science [181]	**ogms**
	Mental Functioning Ontology [182]	**mf**
	Neuro Behavioral Ontology [183]	**nfo**
	Phenotypic quality [184]	**pato**
	Ontology for clinical research and base expertise [7]	**visf**
	Informed Consent Ontology [8]	**ico**

Table 5.2 – Ontology imports.

tinuant leverages BFO expressiveness. A KDDM execution is then expressed as an *occurent*, achieving the objective defined in KDDM specification. The concept of *information object* which is defined as a "piece of information independently how it is concretely realized" is represented separately using the realization (KDD implementation) and its concretization (execution).

The clear definition of terms for classes and properties varies according to the ontology developing community. The OBO Foundry demands a clear and unambiguous definition, using human-readable property annotations, such as rdfs:label (as primary label), iao:ImportedFrom, obo:hasExactSynonym, to annotate synonyms, abbreviations, examples of use, and other information that can help ontology users to understand the context of the term definition. The NeOn Project and the OBO Foundry highlight the importance of making reference to external classes whenever is possible in order to minimize duplication of efforts. OPIS imports and/or extends classes from SSN-O, the OntoDM, and their respective upper and middle-level ontologies (DUL, BFO, OBI, IAO, and SWO).

In addition, OPIS also makes cross-references with classes from external ontologies, as presented in Table 5.2 and described below:

— Environment Ontology (ENVO) [9]: ontology for specifying a wide range of environments relevant to multiple life science disciplines [180].

4. https://github.com/BFO-ontology/BFO (accessed on 26/04/2017)
5. http://www.obofoundry.org/ontology/ro.html (accessed on 26/04/2017)
6. http://purl.obolibrary.org/obo/obi (accessed on 26/04/2017)
7. https://github.com/information-artifact-ontology/IAO/ (accessed on 26/04/2017)
8. https://github.com/openrif/vivo-isf-ontology (accessed on 26/04/2017)
9. https://github.com/ICO-ontology/ICO (accessed on 26/04/2017)
9. http://environmentontology.org/ (accessed on 26/04/2017)

— Ontology for General Medical Science (OGMS) [10]: ontology of entities involved in a clinical encounter, describing the *human being*, *bodily process* and medical disciplines, such as diseases, disorders, patients, so forth [181].
— Mental Functioning Ontology (MF) [11]: ontology for describing human mental functioning, such as belief, cognitive capabilities, personality, mental processes, and so forth. [182].
— Neuro Behavioral Ontology (NBO) [12]: ontology for systematic representation of behavior process and behavioral phenotype [183]. The behavior process is extended from the Gene Ontology (GO) [185] class of biological behavior while the behavior phenotype characterizes behavior and allows classifying its quality as normal or abnormal.
— Phenotypic quality (PATO) [13]: ontology for phenotype quality ontology [184], which provides a framework to represent phenotypes and its qualities. The quality branch of PATO can be used to describe *behaviors* using one or more entities [183].
— Ontology for clinical research and base expertise (VIVO-ISF) [14]: ontology for researchers and research domain based on the open-source project CTSAConnect [15]. It includes classes of project, laboratory, and geographic location, providing a well-defined ontological framework integrated to other OBO Foundry ontologies.
— Informed Consent Ontology (ICO) [16]: ontology for documentations and processes in human subject research involved in informed consent. ICO provides representation for personal information, anonymized data, and informed consent processes.

In order to deal with the extensive number of classes and imports, we followed the Minimum Information to Reference an External Ontology Term (MIREOT) guidelines, created in the context of OBI development [186]. These guidelines provide a method for optimizing the reuse of existing ontology resources, minimizing rework for tasks related to import specific classes. The method consists of gathering the minimum information from the external class, targeting only the class or branch of the ontology relevant to represent the imported class. The OntoFox tool [17] [187] was developed to assist the MIREOT method.

10. https://github.com/OGMS (accessed on 26/04/2017)
11. https://github.com/jannahastings/mental-functioning-ontology (accessed on 26/04/2017)
12. https://github.com/obo-behavior/behavior-ontology (accessed on 26/04/2017)
13. https://github.com/pato-ontology/pato/ (accessed on 26/04/2017)
14. https://github.com/openrif/vivo-isf-ontology (accessed on 26/04/2017)
15. https://github.com/openrif (accessed on 26/04/2017)
16. https://github.com/ICO-ontology/ICO (accessed on 26/04/2017)
17. http://ontofox.hegroup.org/ (accessed on 26/04/2017)

5.4 OPIS – ONTOLOGY FOR PERSONAL INFORMATION ON SENSOR WEB

The ontology contains 69 classes and 5 object properties, 1 data property and it is available at https://github.com/thiagomoreirac/opis (accessed on 26/04/2017). These classes and object properties are aligned to upper-level ontologies, and extended from middle-level and domain ontology classes, as previously described in Table 5.2. Besides that, the equivalence between BFO-based classes and these extensions are provided.

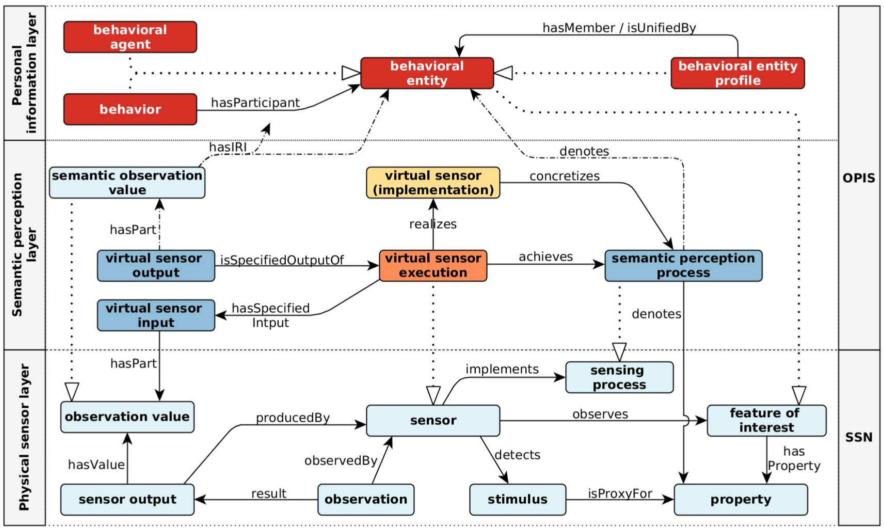

Figure 5.2 – OPIS overview. Red boxes represent behavioral entities. Blue, yellow, and orange boxes represent respectively the specification, implementation and execution representation for KDDM processes according OntoDM. Cyan boxes represent the SSN-O classes.

OPIS is defined in two layers of semantic representation: the Semantic Perception Layer (SPL) and the Personal Information Layer (PIL). Figure 5.2 illustrates in a higher-level how these two layers are arranged among each other and associated with the physical layer constituted by the SSN-O.

In the SPL, the Semantic Perception Process (SPP) is described, along with its implementation (*virtual sensor*), and execution (*virtual sensor execution*). The SPP extends the ssn:sensing and denotes the observed *properties* (ssn:property) and perceived *behavioral entities*. The *virtual sensor execution* is an extension of the ssn:sensor. Instead of *stimulus*, *virtual sensor execution* has observation datasets (*virtual sensor input*) as input and one *semantic observation value* in a *virtual sensor output* as output. This *semantic observation value* ultimately points to an IRI of a semantic representation (class, object property, data property, annotation property) that describe a personal information.

In the PIL, a set of interrelated concepts is defined to represent any feature that may describe or participate in a behavior performed by some *behavioral agent*. Each class is extended from the concept of *behavioral entity* and should be specialized according to the application domain. Three levels of abstractions are defined in PIL: *behavioral feature*, *behavior*, and *behavioral entity profile*. *Behavioral features* are the lowest level of information modeling and are formed by class axioms and object properties that concretely represent personal information, such as information about personal plans, belief, objects, work contexts, and so forth. *Behaviors* correspond to events on which these entities participate. *Behavioral entity profiles* represent higher-level information derived from aggregations of *behavioral entities*.

5.5 BEHAVIORAL MODEL FOR PERSONAL INFORMATION

Personal information is a commonly overloaded term referring both to an individual and information about oneself; as well as to information controlled and owned by someone [20]. As stated in Section 1.3, in the scope of this thesis, *personal information* is defined as *all data, information, and knowledge related to an individual and/or under her control*.

The analysis, classification, integration and management of personal information is a challenging task due to its different aspects and perspectives. Traditionally, personal information annotation and classification were proposed in the context of Personal Information Management (PIM) systems [188]. With the diffusion of the Semantic Web technology and ontologies for knowledge representation and management, *personal information* started to be formalized using OWL encoding. However, since personal information were essentially created and managed using computers, most of the PIM-based ontologies describe personal information as artifacts or media, such as files, documents, images, email etc. More recently, with the advance of pervasive computing and information processing techniques, all sort of personal information can be inferred based on the continuous observation of individuals using sensor data.

The behavioral modeling provided by the Behavior Computing (BC) offers a systematic way of understanding features that explains or predicts behaviors. In BC, *behaviors* refer to actions, operations or events conducted by agents within certain context and environment (virtual or physical ones), focusing on symbolic behaviors that represent these activities into a computational model. The convergence between personal information representation and the recognition of human behaviors is due mainly because of the extensive use of machine learning and data mining techniques to extract information from sensor data. BC not only provides the model and tools to structure, model, represent, simulate, analysis, and use, but also the behavioral feature space to convert *transactional data* and sensor data to *behavioral data*.

5.5.1 Ontologies for Personal Information and Behavioral Recognition

The use of ontologies to represent personal information has been investigated in different contexts, such as PIMs and desktop environments [189, 141]; and behaviors/human activity recognition [190, 191, 192, 174]. Ontologies for personal information on the desktop environment are semantically richer about subjects (agents) and objects (personal information managed during a behavior) while ontologies for human activity recognition tends to focus on the description of behaviors (activities), contexts and locations. Table 5.3 presents a representative set of ontologies for personal information in the context of PIMs, desktop applications, and human activity recognition, and its mapping to the set of *behavioral features* proposed by the BC.

Several concepts are specialized according to the knowledge domain. For example, PIMO [141] describes specific concepts of file system organization, folders, views, and tags used to classify media and artifacts in a computer. On the other hand, behavior recognition approaches have traditionally exploited probabilistic models, sensor types, body positions, human interactions, and environment conditions [197]. For instance, CoDAMoS represents information about user, environment, platform, and service in ambient intelligence. Users can be specified in term of task, profile, role, mood, and preferences. Platform has software, hardware, and environment with location, time, and environmental conditions, such as temperature, pressure, and noise. Service has profile, model, software provider and is used by tasks (and activities).

Practically, both sets of information are related to individuals and the way they interact with the computer, information, or the environment. Smart meters that capture environment conditions, such as CO_2, luminance, and air quality levels, are currently used to optimize temperature regulation in smart building [198]. However, these smart meters can impose serious privacy threats because of the information that can be inferred and predicted about details of a home place, such as a number of occupants, their daily routine, and their devices [199]. The gap between this information and the concept of privacy (i.e., personal information) evidences the inability of current ontologies to represent the association between lower-level data and meaningful person information that is understandable by end-users.

There are still several concept overloads in this subset of personal information approaches. The lack of the adoptions of Linked Data principles [18] or ontology design best practices, such as the alignment with foundational and middle-level ontologies, are one of the main reasons for this problem, and hence limits the interoperability of systems that adopt these ontologies.

In that direction, Khefifi [142] proposes an ontological framework using the concepts of points of views, organizing the reuse of domain-specific ontologies to classify and contextualize personal information.

18. https://www.w3.org/DesignIssues/LinkedData.html (accessed on 26/04/2017)

Behavioral Feature	**CoDAMoS Ontology [193]**	**PiVOn - Context Model Ontology[194]**	**Complex Human Activities Ontology[195]**	**Personal Ontology[196]**	**PIMO[141]**	**Foundational Ontology-based Human Activity Model[192]**
subject	user	user	person - actor	person - organization	party - person - organization - group of persons	user - agent
object		service - object - content	artifact	animal - plan - artifact - document - book - journal - food - substance - natural object	person - organization - social event - document - information element	physical object
place	location	environment - area - space	symbolic location	place - country - city - state - - home	location - building - city - country - room - state -	place
context	environment condition	user situation - event		event	event - social event - meeting	situation
belief - knowledge				education - expertise - ability	knowledge base	
action	task	task	action		task	
goal		goal				
plan		schedule			project	
impact						
constraint				physical characteristics		
status						
associate	role	role - contact	interaction type		class role - role of person	role
time	time	time	time extent			SWRL temporal
behavior	activity		activity - social activity - individual activity -	activity - research - teaching -		activity
profile - pattern	profile	user profile				profile

Table 5.3 – Comparison of approaches for personal information representation based on ontologies

Term	Description	Example	SSN-O mapping	Information Processing
entity	An object or event in the world.	apple	ssn:entity	ssn:featureOfInterest
quality	An inherent observable property of an *entity*.	red	ssn:quality	dul:region
quality-type	A category (or class) of *quality*	color	ssn:quality	ssn:property
percept	A *quality* that has been detected	red	ssn:quality	ssn:observationValue
observer	An agent that executes the *observation-process*	sensor	ssn:sensor	ssn:sensor
perceiver	An agent that executes the *perception-process*	computer	-	**information processing unit**
focus	A *quality-type* whose detection may reduce the *perceptual-theory*	color	ssn:quality	ssn:property
perceptual-theory	A set of *entities* that each explains a set of percepts	{apple, nose}	-	**predictive models**
inheres-in	A relation between a *quality* and an *entity*	red inheres-in apple	ssn:isQualityOf	ssn:isPropertyOf
has-type	A relation between a *quality* and a *quality-type*	red has-type color	-	ssn:hasRegion
observation-process	An act of detecting a *quality* and generating a *perception*	observation-process(red) → red	ssn:observation	ssn:observation
perception-process	An act of inferring a *perceptual-theory* from a set of percepts	perception-process(red)→{apple, nose}	-	**information processing execution**
perception-cycle	An act of minimizing a *perceptual-theory* by focus attention	perception-cycle (...)→{apple}	-	
-	Information quality related to the information processing	{information accuracy, level of confidence}	-	**inference quality**
-	Information about the *perception-process*	{algorithms, parameters}	-	**inference provenance**

Table 5.4 – The original mapping IntelleO x SSN-O from [200] versus our mapping based on information processing concepts.

The framework focus on a formal definition for ontology matching to associate: i) personal information, ii) information type and classification, and iii) personal and informational context. The set of semantic mappings between different contexts, ontology axioms, and information type delivers constitute the user's preference and it is used to infer about personal information and privacy policy conditions. A similar approach for multi-ontology-based PIM is proposed in [201]. Xiao *et al.* [201] define a layered ontology-based framework with annotations, association and navigation capabilities. Both strategies do not define any ontology for personal information, restraining to formalize the ontological framework necessary to annotate and retrieve artifacts.

The representation of personal information in different contexts has not been properly addressed, in particular in situations related to the pervasive computing and the Sensor Web. In the next section, we present the Semantic Perception (SP) paradigm to address the problem of representation between personal information and sensor data.

5.6 THE SEMANTIC PERCEPTION PARADIGM

5.6.1 Information Abstraction and the Semantic Perception

The concept of information abstraction is commonly found in information processing scenarios, referring to the process of generalization and incorporation of knowledge from a low-level context to a high-level context. Sigg *et al.* [202] define information abstraction as *the amount of processing applied to the data and the accumulated information accuracy associated with this process*. This highlights how the data provenance, such as physical and virtual environment contexts, parameters and conditions, are part of the information obtained from data processing. Ganz *et al.* [203] define two level of information abstraction with regard to the sensor-centric and user-centric perspective of the IoT: *data abstraction* and *semantic abstraction*. The former represents atomic and static information obtained by gathering timestamped data from physical sensor streams or its semantic annotation, such as sensor type, capability, accuracy etc. These annotations are commonly represented using the Semantic Sensor Network ontologies such as the SSN-O. The latter corresponds to the inferred information obtained by observing data abstractions and generalizing these observations using logical inference, data mining or machine learning techniques. The result of this process is interpreted according to known patterns and expressed as semantic representations.

Henson *et al.* [200] introduce an ontology-based solution for *semantic abstraction* based on the theories of perception and logical inference. The approach provides a Semantic Perception (SP) process using semantic reasoning and an ontology – IntellegO – that perceives higher-level information based on sensor data. The main ideas underlying this work relies on the cyclic nature of the SP process, background knowledge representation, and the hypothesis test using the sensor data. The perception process is formally modeled into IntellegO, allowing to infer, through OWL reasoning, semantic representations that better interpret a set of low-level observations (sensor data). Henson *et al.* utilize the concept of *perception-cycle* (focus) and *graceful degradation* (abductive inference) to provide the best explanation for the perception process, even when sensor observations are incomplete [68].

In order to provide compatibility to the SSN-O, a mapping between IntellegO and the SSN-O is provided in [200]. This correspondence is presented in Table 5.4 (in columns term, description, example, and SSN-O mapping), along with our mapping to SSN-O and information processing concepts. Our mapping to SSN-O diverges from their original one based on the criteria of using the most specialized SSN-O entities as substitutes for generic terms. For example, instead of specifies ssn:entity as object or event in the world, the ssn:featureOfInterest represents more specifically the same concept.

Some semantic representations not covered by the SSN-O or the IntellegO, such as the *information processing unit*, *predictive model*, *information processing execution*, *inference quality* and *inference provenance*, are a consequence of the increasing usage of information processing to process sensor data. In the next section, we discussed these characteristics and the available semantic representations for them.

5.6.2 Meta-Mining for Semantic Perception and Virtual Sensors

The strong influence of the Sensor Web and the IoT has brought attention to the *connected things* with sensory and information processing capabilities [1], enlarging the boundaries of sources (sensors, social network streaming, databases) and application partitioning [62]. Thus, *virtual sensors* constitute a conceptual layer that concretely implements information processing scenarios, leveraging the Semantic Sensor Network from a physical to a perceptual and cognitive sensing.

However, the term *semantic perception*, which was originally coined in [200], implemented the perceptual model using an ontology-based abductive logic framework. Even if *virtual sensors* implement other information processing scenarios that result in the perception of semantic representations, such as those works classified as CEP [84], the term SP would not be employed correctly.

In this thesis, we retain the semantics of the term *Semantic Perception* (SP) to *represent any information processing scenario that infers high-level information from lower-level data*. However, we extend the scope of its applications beyond the original focus on the use of logical inference to include data mining and machine learning techniques for realizing SP. We use this concept to address the issue of association between sensor data and personal information by designing SP as the new paradigm of sensing for *virtual sensors* that implement perceptual or cognitive sensing. As a consequence, *virtual sensors* are defined as information processing implementations of SPs that produce semantic representations, in particular, those defined as personal information.

Behavior Feature	**Behavioral Entity**	**Realist Mapping BFO-based**	**Descriptive Mapping DUL-based**
s	behavioral agent	$\sqsubseteq$ obi:human being	$\sqsubseteq$ dul:agent
o	actionable object	$\sqsubseteq$ bfo:object	$\sqsubseteq$ dul:object
w	geographic feature	$\equiv$ envo:geographicFeature	$\sqsubseteq$ dul:physicalPlace
	geographic location	$\equiv$ visf:geographic location	$\sqsubseteq$ dul:place
b	belief	$\equiv$ mf:belief	$\sqsubseteq$ dul:concept
a	behavior action	$\sqsubseteq$ ogms:bodily process	$\sqsubseteq$ dul:action
g	behavioral goal	$\sqsubseteq$ iao:objective specification	$\sqsubseteq$ dul:goal
e	context	$\sqsubseteq$ envo:environmentCondition	$\sqsubseteq$ dul:situation
l	behavioral plan	$\sqsubseteq$ iao:planSpecification	$\sqsubseteq$ dul:Plan
u	status	$\sqsubseteq$ pato:processQuality	$\sqsubseteq$ dul:planExecution
t	temporalRegion	$\equiv$ bfo:temporalRegion	$\equiv$ dul:timeInterval

Table 5.5 – Behavioral entity classes origin and mapping

Since SP and *virtual sensors* are the specification and implementation of information processing scenarios, they are representable with OntoDM concepts. This Meta-Mining (MM) ontology provides semantic representations that were found neither in IntellegO nor in the SSN-O to specify *virtual sensors*.

5.7 THE PERSONAL INFORMATION LAYER

According to the SSN-O, *features of interest* can represent an abstraction of real-worldd phenomena, such as persons, events, data, or any measurable thing. SSN-O purposely defines features at this level so domain-specific applications can instantiate them. This domain-agnosticism related to the observed feature of interest and its properties does not permit to classify as personal either these features or the sensing data of the observed properties. In the Personal Information Layer (PIL), we specify a new conceptual layer to represent personal information by extending features of interest into *behavioral entities*.

Behavior entity is the semantic concept that represents the BC concept of *behavioral feature* based on the SSN-O concept of *feature of interest* ($\sqsubseteq$ *ssn*:Feature-OfInterest), as defined in Vector 3.1. As *features of interest*, *behavioral entities* can be observed using sensors and *virtual sensors* where the sensor output represents the ssn:observationValue of a specific ssn:property, and the *virtual sensor output* represents the *semantic observation value* of *behavioral entity* representations. We classify *behavioral entities* in *behavioral entity classes* and *behavioral entity properties*. The former is related to entities that exists from observable individuals, and it is defined as:

$$\gamma_c = \{s, o, w, b, a, g, e, l, u, t\} \tag{5.1}$$

The latter consists of properties that are observable from associative relations or state transitions, defined as follow:

$$\gamma_p = \{f, c, m\} \tag{5.2}$$

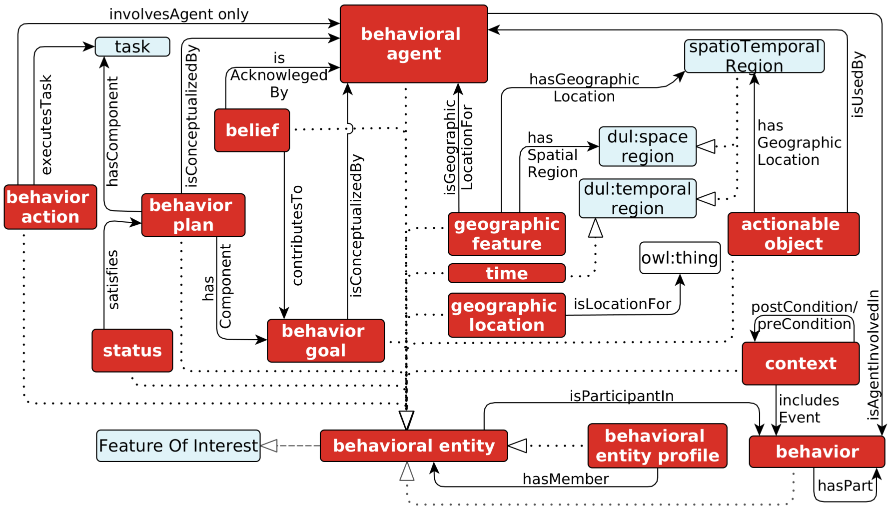

Figure 5.3 – *Behavioral entity classes* and their relationships. Cyan boxes represent SSN-O/DUL entities. Dotted arrows represent the subsumption relationship.

Figure 5.3 presents the *behavioral entity classes* (γ_c) and their relationships. *Behavior entities* is a superclass of all *behavioral features*, including *behavior* itself and pattern vectors of behaviors (*behavioral entity profile*). Aiming to represent *behavioral entities* as *personal information*, we conceive *behavioral entities* as subclasses of ssn:featureOfInterest and the ICO class of *personal information*.

```
BehavioralEntity ⊑ ssn:featureOfInterest ⊓
                 ⊑ ico:personalInformation
```

For the definition of γ_c entities, we observed DUL and BFO concepts, aiming to represent them from a descriptive and realist perspective. The terminology was inspired or incorporated from external ontologies indexed in the OBO Foundry (BFO-based) and from DUL entities. Consequently, two mapping were generated from this process, resulting in the terminology, imports, and reuses of *behavioral entity classes* presented in Table 5.5. The behavior feature *place* is represented in two senses: *physical location* and *geographic location*, contemplating respectively the physical and descriptive view. The remainder of the γ_c entities has the same semantics of *behavioral features*.

Object Property	**Realist Mapping BFO-based**	**Descriptive Mapping DUL-based**
hasGeographicLocation	≡ visf:hasGeographicLocation	⊑ dul:locationFor
isUsedIn	⊑ ro:functionallyRelatedTo	–
hasSpatialRegion	⊑ ro:locatedIn	⊑ dul:hasRegion
contributesTo	–	⊑ dul:isConceptUsedIn
isAcknowledgedBy	–	⊑ dul:isConceptualizedBy

Table 5.6 – Set of behavioral entity class properties (descriptive perspective)

Any information produced by or related to an individual is personal. These relationships are represented as object properties restrictions in *behavioral entity classes*. DUL provides the most part of these object properties. However, five new object properties were not found in DUL nor in BFO-based ontologies to describe some relationships between *behavioral agent* and γ_c entities are presented in Table 5.6. However, as highlighted in the table, differently from the γ_c entities, not all correspondences in DUL and OBO Foundry ontologies are found for the proposed object properties. The object property *hasGeographicLocation* defines a *geographic location* where a physical object or place exists in a specific time. The *isUsedBy* property represents the utility of an *actionable object* for a *behavioral agent*. Next, *hasSpatialRegion* defines a spatial region for a physical object or place. *ContributesTo* represents the influence of a *belief* in *behavioral goal*. Lastly, *isAcknowledgedBy* property corresponds to the mental capacity of a *behavioral agent* to know, reason, assume, believe, or understand a specific *belief*.

Axiom	**Expression**
geographic feature	$\sqsubseteq \exists$ hasSpatialRegion dul:spaceRegion
	$\sqsubseteq \exists \leqslant 1$ isGeographicLocationOf behavioralAgent
geographic location	$\sqsubseteq \exists \leqslant 1$ locationFor owl:Thing
behavioral agent	$\sqsubseteq \forall$ dul:isAgentInvolvedIn behavior
actionable object	$\sqsubseteq \exists$ hasGeographic location geographicFeature
	$\sqsubseteq \forall$ isUsedBy behavioralAgent
belief	$\sqsubseteq \exists$ contributesTo BehavioralGoal
	$\sqsubseteq \forall$ isAcknowledgedBy behavioralAgent
behavior action	$\sqsubseteq \forall$ dul:involvesAgent behavioralAgent
	$\sqsubseteq \exists$ dul:executes dul:task
behavioral goal	$\sqsubseteq \forall$ isConceptualizedBy behavioralAgent
behavioral plan	$\sqsubseteq \exists$ dul:hasComponent dul:task
	$\sqsubseteq \exists$ dul:hasComponent behavioral goal
	$\sqsubseteq \forall$ isConceptualizedBy behavioralAgent
status	$\sqsubseteq \forall$ dul:satisfies behavioral plan
context	$\sqsubseteq \exists$ dul:hasPostCondition context
	$\sqsubseteq \exists$ dul:hasPreCondition context
	$\sqsubseteq \forall$ dul:includesEvent behavior
impact	$\sqsubseteq \exists \leqslant 2$ dul:isSettingFor behavioral

Table 5.7 – Semantic expressions for behavioral entity class restriction

Based on these object properties, we present in Table 5.7 the initial and extensible set of semantic expressions that defines *intensionally* γ_c entities based on the relationship between *behavioral agent* and the rest of γ_c entities. Firstly, we define the *geographic feature* as a spatial entity that has *hasSpatial* regions. Next, we related the spatiotemporal existence of physical object entities *behavioral agent* and *actionable object* to geographical location (*hasGeographicLocation*) using the *geographic feature*. *Geographic features* were additionally described as being a geographic location for at least a *behavioral agent*. The descriptive entity for place, *geographic location*, defined as location for (dul:locationFor) at least one *thing*.

Behavior Feature	Behavioral Entity	Realist Mapping BFO-based	Descriptive Mapping DUL-based
f	impact	$\sqsubseteq$ ro:causalRelationBetween-Processes	$\sqsubseteq$ dul:precedes
c	constraint	$\sqsubseteq$ ro:causesCondition	$\sqsubseteq$ dul:isConstraintFor
m	associate	$\sqsubseteq$ ro:temporallyRelatedTo	$\sqsubseteq$ dul:associatedWith

Table 5.8 – Relational behavioral entities

The involvement (dul:isAgentInvolvedIn) and its inverse property (dul:involvesAgent) of the *behavioral agent* to the *behavior* is required and becomes the entailment that associates the observation and the behavior abstraction level.

The *actionable object* is defined as any object (dul:object) used by some *behavioral agent*, i.e., any physical, agent, or social object that has a utility in the context of a behavior observation. This object can participate and imposes restrictions to the *behavior*. For instance, the object "bike" can restrict the mobility behavior to bike lanes and the agent's speed.

The notion of belief as mental representation is precedent and necessary in shaping our action and contributes to determine our behavior strictly depending on the pursuit goals [204]. Therefore, we extended the DUL class of concept (dul:concept) to represent *beliefs* which, in turn, can be used to define, characterize, parametrize, cover social objects (dul:socialObjects). The mental function performed by a *behavioral agent* is represented in using the property *isAcknowledgedBy*.

Action is an event realized by one or more agents aiming to accomplish a goal or follow a plan. Thus, we specify *behavior action* as a specialization of DUL class of action (dul:action) on which participates (dul:involvesAgent) any *behavioral agent*. The concept of dul:action is conceptualized as a task execution. We specialize hence the DUL class of plan (dul:plan) to described *behavioral plan* as some social object composed of tasks and goals and conceptualized by *behavioral agents*. In addition, DUL defines properties that permit specifying precedence and *part-hood* of actions, tasks, and plans. Then, the concept of *status* is a process quality that expresses a state of the plan execution using a DUL class of description (dul:description).

Lastly, the concept *context* defines a condition of an environment that may have post condition (dul:hasPostCondition) or pre-condition (dul:hasPreCondition), and it is involved in some *behavior*.

The set of *behavior entity properties* γ_p is presented in Table 5.8. These entities are conceived as properties because of its relational nature. The concept of *impact* property captures the causality between two or more *behaviors* or precedence in terms of that causality, extending the DUL property of precedence (dul:precedes) and the OBO RO property of causal relation between processes (ro:causalRelationBetweenProcesses). The concept of *constraint* has a causality between a *behavior* and a *behavioral entity class*, extending the DUL property of constraint relationship (dul:isConstraintFor) and the OBO RO property of condition causality (ro:causesCondition). Lastly, the *associate* property represents the relationship that *behavioral entities* have with each other, being an extension of DUL property of association (dul:associatedWith) and OBO RO property of temporal relation (ro:temporallyRelatedTo).

A *behavior* is an event which concept is defined by the set of *behavioral entities* that participates in the event. Since *behaviors* can be composed of other *behaviors* ($\sqsubseteq \exists$ *ro*:hasPart behavior), the level of granularity of *behaviors* can also be represented semantically. Figure 5.4 depicts an example of behavior granularity that can be represented using OPIS. Besides that, *behavioral entities* that have a temporal aspect

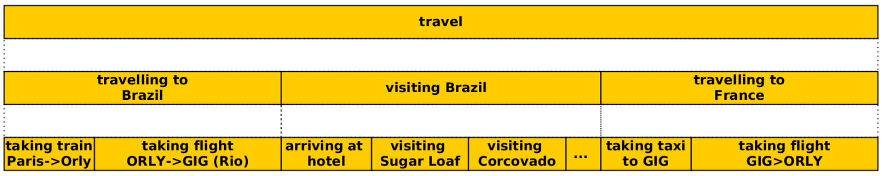

Figure 5.4 – Example of behavior hierarchy

of occurrence can be integrated or derived, changing the representational abstraction level. For instance, the spatial extension of an observed *geographic feature* or the average wind speed associated with a moving agent can change substantially if measured during one second or during one hour.

In particular, *behaviors* can be arranged hierarchically in a tree-like structure which increases the level of detail toward its leaves. Since it is possible to scrutinize and observe *behaviors*, its abstraction level is logically extended to *behavioral entities*.

Ultimately, a behavior pattern can be represented in terms of a sequence of *behaviors* ($\overrightarrow{\tau}$) . We define *behavioral entity profile* as a special *behavioral entity* that represents a type of collection and a data mining generalization about *behavioral entities* that shares certain characteristics and is unified by a pattern.

```
BehavioralPattern ⊑ (dul:typeCollection ⊓
                    ∀ dul:isUnifiedBy dul:pattern) ⊓
                    ⊑ odm:generalization
```

This social object provides an effective way to describe generalizations found in *behavioral entities* and their properties. Its descriptive capability based on DUL allows describing the pattern used to unify such collection. For instance, a sportive profile can represent a recurring pattern of running (*behavior*) during a period of a month. Besides that, it can be used to represent data abstraction and patterns, using the concept of odm:generalization.

Subsumption allows modeling *behavioral entity extensionally*, i.e., restricting the class membership by conceptual definition, and *intensionally*, i.e., defining restrictions to object properties, cardinalities, and so forth.

5.8 THE SEMANTIC PERCEPTION LAYER

The Semantic Perception paradigm describes a *virtual sensor* in terms of its input, information processing, and semantic observations. The information processing, its specification, implementation, and execution is described in the Semantic Perception Layer (SPL), as presented in Figure 5.5 and formally defined in Table 5.9. The three levels of representation are grouped around three main aspects: a) semantic perception process specification; b) virtual sensor - implementation and execution; and c) virtual sensor data and dataset specification.

5.8.1 Semantic Perception Process Specification

The OntoDM specification layer provides concepts to describe the information processing scenario (scenario) and its process in terms of algorithms, objectives, and data specifications. Figure 5.6 isolates the entities that are used to specific the Semantic Perception Process (SPP).

OntoDM defines the concept of scenario (odm:scenario) that specifies a sequence of data processing algorithms with inputs and outputs. The SPP is thus defined as a specialization of odm:scenario and ssn:sensing, representing the process which objective is a Semantic Perception (SP). The SPP is composed of (*hasPart*) a *semantic perception objective* and a sequence of algorithms (obi:algorithm), such as data processing algorithm, data mining algorithm, and evaluation algorithm; each of them formed by their respective objective specification. It is important to note that OntoDM specializes data mining algorithms and their structures, offering a richer representation of those information processing techniques not presented here for brevity matters.

Each algorithm has an objective specification which contains data specifications and descriptive information about the objective. OntoDM defines two types of data specification: *output data specification* and *descriptive data specification*. Both are about some odm:datatype and has mapping specifications (odm:mappingSpecification), which concretely associate the data to its data type. In OPIS, SPP are restricted to have algorithms which objective specification contains data specifications associated with observed property and perceived semantic representation. Therefore, the *output data specification* in OPIS is composed of some *semantic mapping specification* and is about some *semantic perception datatype*. The *descriptive data specification* in OPIS is formed by some *property mapping specifications* or *semantic mapping specifications*, which allows specifying input using both the SSO pattern and the SP paradigm. *Property mapping specification* and *semantic mapping specification* extend the OntoDM concept of odm:mappingSpecification to denote, respectively, ssn:property and semantic representation. Datatypes are detailed in the subsection 5.8.3.

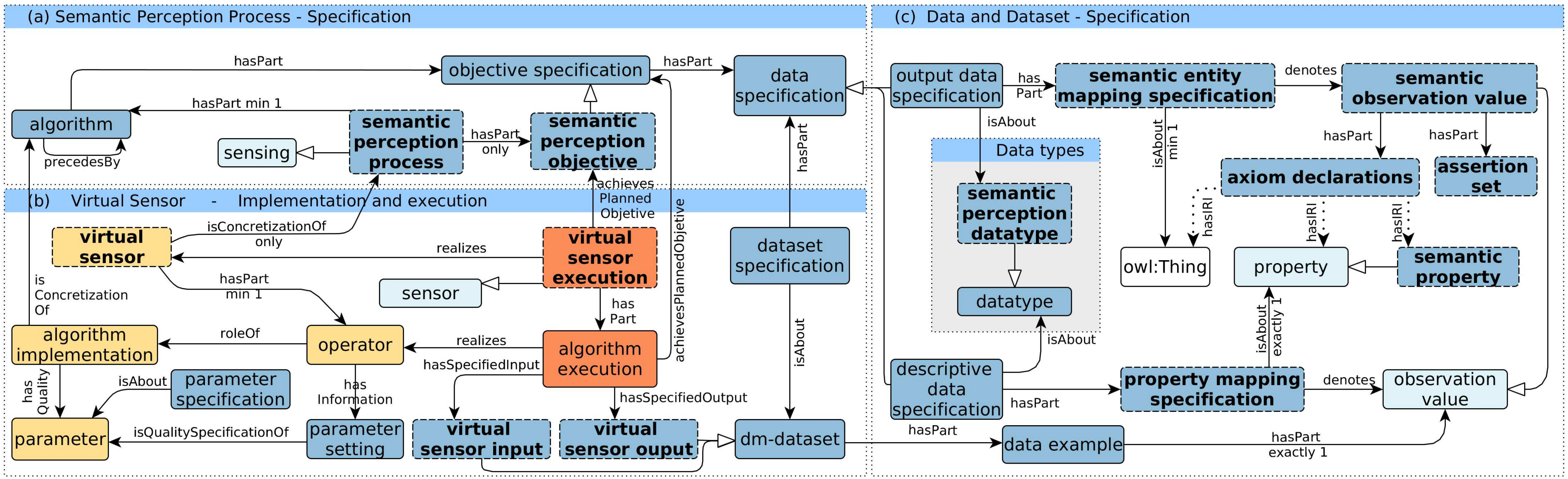

Figure 5.5 – Semantic Perception Layer. OntoDM concepts of specification, implementation and execution levels are represented respectively in blue, yellow, and orange. OPIS concepts are highlighted in bold and dotted line boxes in the same color of the extended concepts. SSN-O is represented in cyan.

Axiom	**Expression**
semantic perception objective	$\sqsubseteq$ iao:objectiveSpecification
semantic perception process	$\sqsubseteq$ odm:scenario $\sqsubseteq$ ssn:sensing $\sqsubseteq$ $\exists$ ro:hasPart (iao:algorithm $\sqcap$ $\exists$ ro:hasPart (iao:objectiveSpecification $\sqcap$ $\exists$ ro:hasPart (odm:outputDataSpecification $\sqcap$ $\exists$ iao:isAbout SemanticPerceptionDatatype $\sqcap$ $\exists$ ro:hasPart semanticMappingSpecification) $\sqcup$ (odm:descriptiveDataSpecification $\sqcap$ $\exists$ iao:isAbout odm:datatype $\sqcap$ $\exists$ ro:hasPart (propertyMappingSpecification $\sqcup$ semanticMappingSpecification)))) $\sqsubseteq$ $\exists$ ro:hasPart SemanticPerceptionObjective
virtual sensor	$\sqsubseteq$ odm:workflow $\sqsubseteq$ dul:informationObject $\sqsubseteq$ $\forall$ obi:isConcretizationOf SemanticPerceptionProcess $\sqsubseteq$ $\exists$ ro:hasPart (odm:operator $\sqcap$ $\exists$ ro:hasRole (odm:algorithmImplementation $\sqcap$ $\exists$ obi:isConcretizationOf (iao:algorithm $\exists$ $\sqcap$ ro:partOf SemanticPerceptionProcess)))
virtual sensor execution	$\sqsubseteq$ odm:workflowExecution $\sqcap$ $\exists$ ro:hasAgent obi:computer $\sqsubseteq$ ssn:sensor $\sqsubseteq$ $\forall$ odm:realizes virtualSensor $\sqsubseteq$ $\forall$ obi:achievePlannedProcess semanticPerceptionObjective $\sqsubseteq$ $\forall$ ro:hasPart (odm:algorithmExecution $\sqcap$ $\exists$ ro:realizes (odm:operator $\sqcap$ $\exists$ ro:partOf virtualSensor) $\sqcap$ $\forall$ obi:hasSpecifiedInput VirtualSensorInput $\sqcap$ $\forall$ obi:hasSpecifiedOutput VirtualSensorOutput)
virtual sensor input	$\sqsubseteq$ odm:DM-dataset $\sqcap$ $\exists$ ro:hasPart (odm:dataExample $\sqcap$ $\exists$ ro:hasPart ObservationValue)
virtual sensor output	$\sqsubseteq$ odm:DM-dataset $\sqcap$ $\exists$ ro:hasPart (odm:dataExample $\sqcap$ $\exists$ ro:hasPart SemanticObservationValue)
property mapping specification	$\sqsubseteq$ odm:mappingSpecification $\sqsubseteq$ $\exists = 1$ iao:isAbout ssn:property $\sqsubseteq$ $\exists$ iao:denotes ssn:observationValue
behavioral entity mapping specification	$\sqsubseteq$ odm:mappingSpecification $\sqsubseteq$ $\exists \leqslant 1$ iao:isAbout owl:Thing $\sqsubseteq$ $\exists$ iao:denotes semanticObservationValue

Table 5.9 – Semantic Perception Layer (SPL) entities

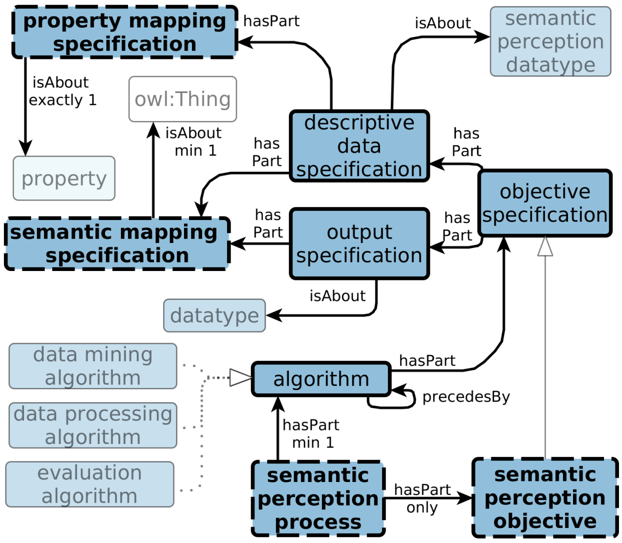

Figure 5.6 – Semantic perception process specification. OPIS classes in bold. Central concepts in thicker line boxes.

5.8.2 Virtual Sensor Implementation and Execution

The association of the implementation and execution of SPP to its specification is realized by object properties between these three endurance representations, such as *isConcretizationOf*, *achievePlannedObjects* etc. The OntoDM implementation layer provides concepts to describe the information processing implementation in terms of algorithm implementations, parameter specifications, parameter settings, and operators. The OntoDM execution layer describes the process by input and output representations used during the information processing execution. Figure 5.7 focus on the implementation and execution representations, adding contextual entities from the SPP specification.

The concept of *virtual sensor* is extended from the OntoDM class of workflow (odm:workflow), and the DUL concept of information object, and must concretize a SPP. Each odm:workflow can be composed of algorithm implementations (odm:algorithmImplementation) and their *operators* (odm:operator). An odm:algorithmImplementation concretizes an iao:algorithm in the same way that odm:workflow concretizes an odm:scenario. Therefore, a *virtual sensor* concretizes a SPP and its odm:algorithmImplementations concretizes iao:algorithms that are part of this SPP.

An odm:operator plays a role to represent an odm:algorithmImplementation that has defined parameter settings (odm:parameterSetting). Parameters (odm:parameter) are optionally related to odm:algorithmImplementations. They have a specification (odm:parameterSpecification) and act as quality specification representation for odm:operators. For example, data mining algorithm implementations of specific software toolkits, such as Weka, can be represented as algorithm implementations along with their parameters ($\sqsubseteq \exists$ *obi*:hasQuality odm:parameter) which are used to adjust the algorithm execution. In addition, these parameters can have

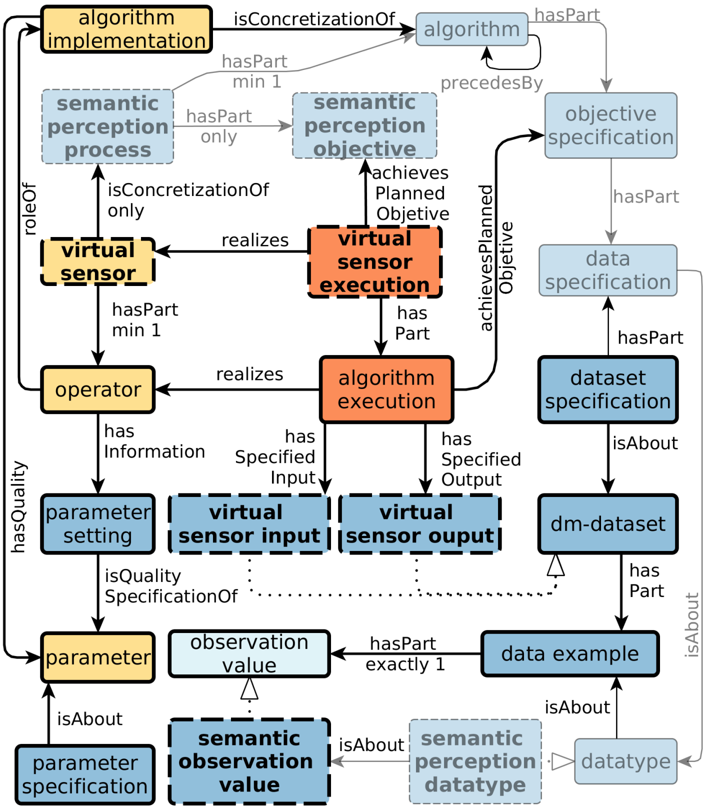

Figure 5.7 – Virtual sensor implementation and execution. OPIS classes in bold. Central concepts in thicker line boxes. Dotted line arrows represent subsumption relationship.

specification (odm:parameterSpecification) to describe format and an identifier for an odm:parameter ($\sqsubseteq \exists$ *iao*:isAbout odm:parameter).

At the execution level, the *virtual sensor execution* extends the OntoDM concept of workflow execution (odm:workflowExecution) and the ssn:sensor, which depends (ro:hasAgent) of a computer (obi:computer) to be executed. It must realize (bfo:realizes) some *virtual sensor* and must achieve (obi:achievesPlannedObjective) a SPP. The OntoDM concept of algorithm execution (odm:algorithmExecution) represents the realization of an odm:operator during the execution of a virtual sensor (*runtime*). In addition, *Virtual sensor execution* has specified input (*virtual sensor input*) and output (*virtual sensor output*) that are used and produced by the *virtual sensor*.

From information discovery perspective, *semantic perception process* is an obi:scenario composed of a sequence of algorithms. This part-hood is reflected into its implementation (*virtual sensor*) and execution (*virtual sensor execution*) structure. For example, if a given *semantic perception process* has specified algorithms A and B, then the *virtual sensor* and *virtual sensor execution* related to this *semantic perception process* will also have respectively sequentially odm:algorithmImplementations A_i and B_i, and odm:algorithmExecutions A_e and B_e.

Virtual sensor input and *virtual sensor output* are specified as extensions of the OntoDM concept of dataset (odm:DM-dataset), formed by data examples (odm:dataExamples) that contain ssn:observationValue and *semantic observation values* respectively. SSN-O defines *observation value* as a region for the sensor output, which is captured during one observation. Sensors produce one data example per observation, which is specified according to the observed feature, property, stimulus, sensor, sensing process, sampling time and so forth. In OPIS, the SSN-O concept of observation (ssn:observation) for *virtual sensors* is represented by the OntoDM concept of mapping specification (*odm:mappingSpecification*) which plays a similar role, associating data specification, data type, and data example. OPIS also specializes the ssn:observationValue to represent *semantic observation values* that contain ssn:observationValue and semantic representations. As a consequence, *virtual sensor input* can be composed of ssn:observationValues or *semantic observation values*. The structure of the *semantic observation value* is described in details in the following subsection.

5.8.3 Virtual Sensor Dataset

Data example are extensions of IAO concept of data item (iao:dataItem) that form an odm:DM-dataset and can represent a single datum, set of discrete, tuple of specific data type, or more a complex structure. Data examples are OntoDM representation for observation values and can be specified according to a data specification (iao:dataSpecification). Conversely, SSN-O define neither the ssn:observationValue formats nor the units of measurements. The OntoDM and the SSN-O designs for observation values and data examples are thus complementary and compatible.

However, the definition of data examples as a set of ssn:observationValues produced by sensors do not address completely the problem of expressing semantic representations as a result of the information processing. Neither odm:datatype nor odm:mappingSpecification, that are part of the data specification that describes odm:DM-dataset, can express this information. OPIS extends the odm:mappingSpecification, the ssn:observationValue, and the odm:datatype to express semantic representation in the algorithm data specification, observation value, and data type. Figure 5.8 presents the concepts used to specify data and datasets, and its association to semantic representations and data types.

PROPERTY AND SEMANTIC MAPPING SPECIFICATION As previously introduced, the odm:mappingSpecification represents the concrete association specified in the odm:dataSpecification between the odm:datatype (specification) and the odm:dataExample (execution). In the OPIS, the odm:mappingSpecification is specialized as *property mapping specification* and *semantic mapping specification*. The former represents the concrete relationship among the descriptive data specification (odm:descriptiveDataSpecification), the observed ssn:property, and the ssn:observationValue. The latter represents the relationship

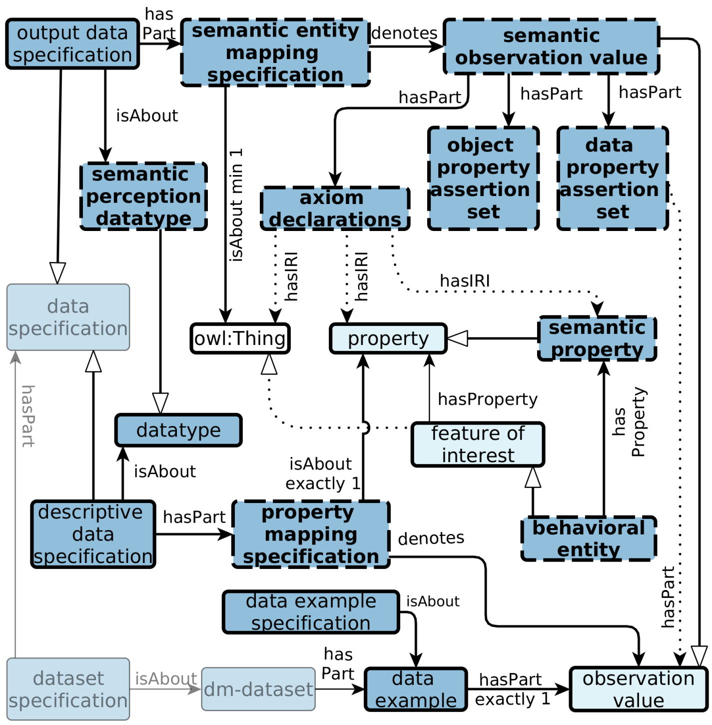

Figure 5.8 – Data and Dataset Specification. OPIS classes in bold. Central concepts in thicker line boxes. Dotted line arrows represent indirect subsumption relationship.

among the output data specification (odm:dataSpecification), the perceived semantic representation (owl:thing), and the *semantic observation value* (ontology axiom, ontology assertion, ssn:observationValue).

The odm:outputDataSpecification and odm:descriptiveDataSpecification are specialized from the IAO concept of data specification (iao:dataSpecification) in OntoDM to differ between output data, its description part, and input data. In that direction, OPIS incorporates odm:descriptiveDataSpecification to specify *virtual sensor input*, linking its odm:mappingSpecification to ssn:property whose observation values compose the data examples of *virtual sensor input*. Similarly, OPIS includes odm:outputDataSpecification to define *virtual sensor output*, associating its odm:mappingSpecification to a semantic representation (owl:Thing) whose instances and classes are part of the data examples of the *virtual sensor output*. Data examples can be also specified according to a *data example specification* that defines some of its quality aspects.

SEMANTIC OBSERVATION VALUE According to SSN-O, ssn:observationValues represent abstract regions that specify sensor output structure and units. SSN-O leaves this definition to the application domain. Complementarily, OntoDM provides representation for data specification and data types. This capacity of OntoDM is used to offer semantic representations to infer automatically about information processing techniques and data compatibility between them.

In OPIS, we propose to bridge the semantic representation by specializing ssn:observationValue and odm:datatype toward the concept of SP. *Virtual sensors* differ mainly from ssn:sensor because of its capacity to produce semantic representation along with observations. Practically, instead of outputting a description or data mining generalization from sensor data, *virtual sensors* are capable of mapping this output as a class assertion, object property assertion, or data property assertion. For the matter of structure, SP can comprise TBox/RBox, ABox, and data properties. The TBox/RBox corresponds to the ontology axiom declarations that will be asserted, ABox refers to the instances of those declarations in class assertions, object property assertions, and data properties. Figure 5.9 presents the *semantic observation value* structure, composed of *axiom declarations* and *assertion set*.

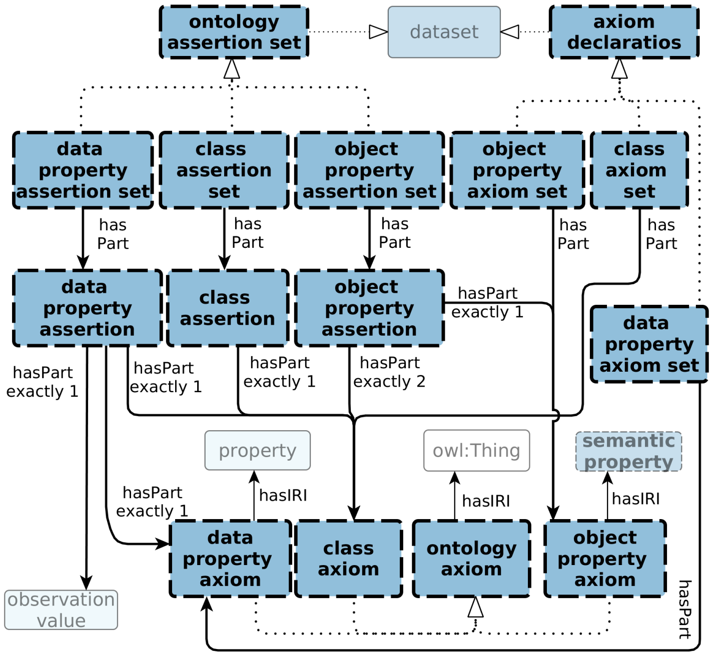

Figure 5.9 – Semantic Observation Value Structure. Central concepts in thicker line boxes. Dotted line arrow represent subsumption relationship.

OWL2 does not provide native support for object property assertions to refer to ontology axioms. For example, be an object property P, which range is A and domain is B, and ClassAssertion(:B :b). It is not possible to assert ObjectPropertyAssertion(:P :b :A). There are few possible solutions for this problem. We propose to represent ontology axioms and assertions which have annotation properties that refer to IRIs in order to intermediate the reference between Abox and TBox/RBox entities. For this purpose, OPIS extends iao:dataItem in two representations: *ontology axiom* and *axiom assertion*.

The first represents *ontology axioms*, such as *class axiom*, *object property axiom*, and *data property axiom*; and has an annotation *hasIRI* to any ontology axiom (owl:Thing). Each *ontology axiom* extension can refer to any semantic representation of an ontology using the annotation property *hasIRI*, in particular to axioms defined in OPIS. *hasIRI* is defined with a datatype xsd:anyURI.

Class axioms can refer to any ontology axiom (owl:Thing), *object property axiom* can make reference to *semantic* *property*, and *data property axiom* can refer to ssn:property. This mapping concretely defines how the *feature-property* perspective from SSN-O is represented in OPIS. In SSN-O, ssn:property is an observable property that results in an observation value ultimately. OPIS represent this perception as *data property axioms* and associate ssn:observationValues to *data property assertions*. However, ssn:property does not comprise explicitly properties observed in semantic representation. OPIS defines thus *semantic property* as observable qualities in a class axiom, in particular, the restrictions defined by *object property axioms* in *class axioms*. It is important to remark that OPIS definition of *data property assertion* allows a complex data type value in the triple (*class axiom*, *data property axiom*, ssn:observationValue).

SEMANTIC PERCEPTION DATATYPE OntoDM has an extensive datatype representation, from primitive datatypes, such as real and characters, to aggregate datatypes, such as classes, pointers, and subtypes. Therefore, the data structure of ssn:observationValue or *semantic observation value*, produced by a *virtual sensor*, can be specified by reusing or extending one of the OntoDM datatypes.

Traditionally, information processing output primitive or aggregated datatypes. OntoDM defines several types of information processing outputs, from the simplest primitive result, such as a real value, to complex data mining results, such as binary classification dataset, multi-target multi-class classification dataset, tree-based hierarchical classification dataset.

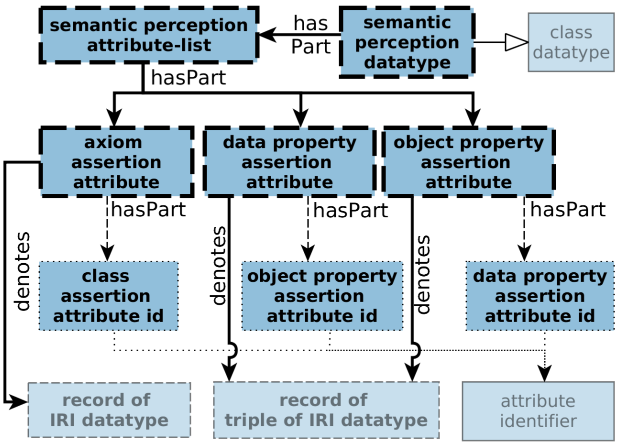

Figure 5.10 – Semantic perception datatype specification. OPIS classes in bold. Central concepts in thicker line boxes. Dotted line boxes represent class instances. Dotted line arrow represents instanceOf (rdf:type).

In order to provide a semantic representation for SP that can involve such complexity, OPIS contain a datatype specification – *semantic perception datatype* – that extends the OntoDM concept of the class datatype (odm:classDatatype), as depicted in Figure 5.10.

The class is defined with a *semantic perception attribute-list* that contains three attributes: *axioms assertion attribute*, *object property assertion* *attribute*, and *data property assertion attribute*. Each of these attributes denotes a datatype and contains an attribute identifier: *class assertion attribute id*, *object property assertion attribute id*, and *data property assertion attribute id*.

In order to represent axioms, we defined *IRI datatype* as an extension of character datatype (odm:characterDatatype) that have an *IRI value space*. Figure 5.11 presents the datatypes used to specify an IRI (*IRI datatype*), a set of IRIs (*record of IRI datype*), RDF triple (*object property field component* and *data property field component*), and a set of RDF triples (*record of triple of IRI datatype*). We define *IRI datatype* in the same value space of OWL2, however, further specialization to restrict the base of IRIs can be achieved using the *IRI base datatype* to specify subtypes of *IRI datatype* (a subset of IRI, for instance).

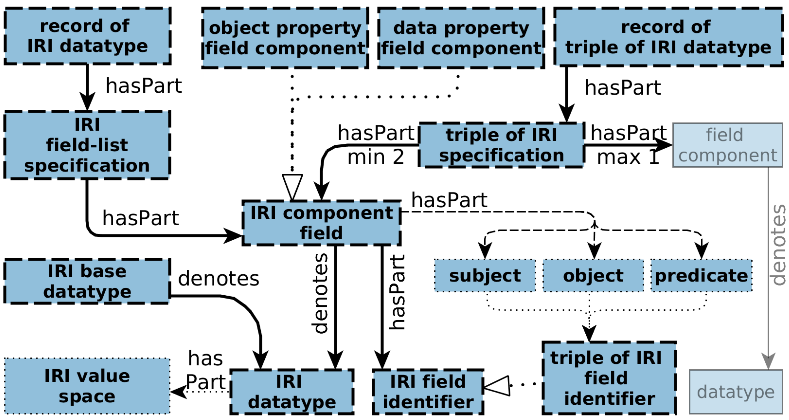

Figure 5.11 – IRI Datatypes. OPIS classes in bold. Central concepts in thicker line boxes. Dotted line boxes represent class instances. Dotted line arrow represents instanceOf (rdf:type).

Two types of datatypes are defined to be used in the *semantic perception datatype*: *record of IRI datatype* and *record of triple of IRI datatype*. The former is an extension of record of character (odm:recordOfCharacter) and has an *IRI field-list specification* that contains exactly one *IRI field component*. The latter is an extension of record of primitives (odm:recordOfPrimitives) and has a *triple of IRI specification* that contains exactly three field components (odm:fieldComponent) on which a minimum of two are *IRI field component*. These *IRI field components* are specialized to represent *object property field component* and *data property field component* and have one of the *triple of IRI field identifiers*: *subject*, *object*, and *predicate*.

Figure 5.12 presents an example of specification for ssn:observationValue using the concepts of record of real (odm:recordOfRealDatatype), an aggregate datatype, to represent the WGS84 standard [19]. From the SSN-O perspective, an observed *geolocation property* from an *agent* is associated with a *property mapping specification*. Each observation generated has one sensor output associated, which, in turn, has a *geolocation observation value* that is defined in the WGS84 specification.

The *WGS84 datatype* has a primitive field-list that specifies two fields of real type: latitude and longitude (instances of field identifier).

19. http://earth-info.nga.mil/GandG/wgs84/ (accessed on 26/04/2017)

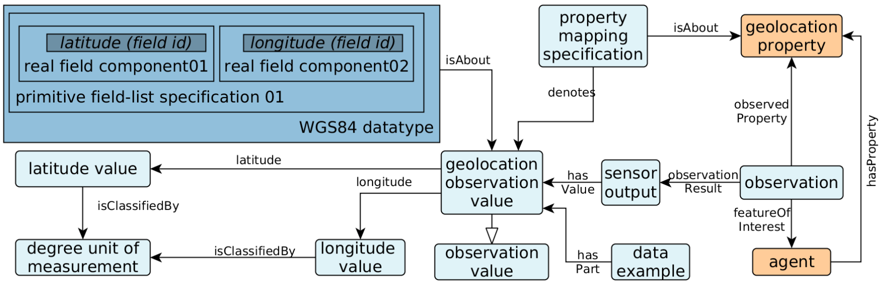

Figure 5.12 – An example of WGS84 data type specification for geographic location (property) observation value. Data type specification represented in dark blue. Observation value classes represented in light blue. Observed property and feature represented in orange.

Therefore, the specialization of the *geolocation observation value* defines two object properties (latitude and longitude) which ranges are the respective observed latitude and longitude values, and whose data properties must match the same component datatype (real datatype). In addition, object properties (latitude and longitude) from the *geolocation observation value* match the field component identifiers.

5.9 ONTOLOGY COMPETENCE

OPIS is a modular ontology, which allows using its conceptual layers separately. In this section, examples and a use case are provided to illustrate the use and instantiation of OPIS. In the first part, three examples from published papers are represented to demonstrate how different applications can use the same conceptual framework provided in the PIL. In the second part, an algorithm for human activity perception is explained and represented using OPIS. Finally, the competence questions (Table 5.1) are answered using the SPARQL-DL notation [20].

5.9.1 Examples

The first case is based on the work of Liu *et al.* [205] that exploits sportive activities using a machine learning time series *shapelets* technique to detect atomic and multi-layered human activity, such as sitting, walking, jumping, jogging, and dribbling; from accelerometer and gyroscope sensors embedded in mobile phones. The second use case is a medical cloud-based approach proposed by Mohammad Forkan *et al.* [206] for disease symptoms detection based on vital signs captured from body sensors that read pulse rate and blood pressure. The third case is based on the work of Krishnan *et al.* [96] that use machine learning techniques to infer about in-door activities from smart-home sensors.

20. http://www.derivo.de/en/resources/sparql-dl-api/ (accessed on 26/04/2017)

The second part of the examples shows the usage of SPL concepts to represent the data mining and machine learning techniques used to recognize disease symptoms, as defined in [206].

PERSONAL INFORMATION REPRESENTATION The Personal Information Layer (PIL) of OPIS offers a conceptual framework to associate several types of information to individuals. Three different scenarios are depicted in Figure 5.13, as highlighted in different colors, showing how the *behavioral entities* can associate information to individuals.

In the first use case (Krishnan *et al.* [96]), in-door activities of elderly residents were detected using smart meters that sense luminance, motion, and doors. To describe in-door environments, the *geographic location* is specialized in home place and then extended to represent a living room, kitchen, room, bathroom etc. The sensors that detect human interaction with the environment are associated with these *geographic locations* as properties. The obtained activities are represented as an extension of behavior / activity. The data mining classification algorithms are capable of detecting bathing, eat, enter home, leave home, cook, take medicine, relax, sleep activities.

In the second use case (Liu *et al.* [205]), smartphones were installed in different parts of the body to collect gyroscope and accelerometer sensor data in order to detect atomic and complex movements during the sport practice. Therefore, the concept *behavior* is extended to describe activities, which, in turn, is extended to represent the activities of bouncing, walking, dribbling, throwing, and jumping. The jump-shot activity is defined as composed movement of jumping and throwing. The class person is extended from *behavioral agent* to represent the body members (limb and arm) used to observe the property acceleration and rotation.

Lastly, the ViSiBiD approach proposed by Mohammad Forkan *et al.* [206] detects hypoxia, tachycardia, and hypertensive symptoms based on pulse rate and blood pressure. For this purpose, pulse rate, diastolic blood pressure, and systolic blood pressure are represented using the Foundational Model of Anatomy [21] and associated as properties to *behavioral agents* / persons. The disease symptoms are represented importing classes from the Human Disease Ontology [22] and are associated with *behavioral agents* / person through the *doid:hasSymptons*.

The ontological framework provided by OPIS allows associating information that was not directly related to individuals. Smart meter sensors, for instance, can be considered personal when its instance are defined along with the assertion *isLocationFor* between its feature (home place, living room, etc) and a person.

SEMANTIC PERCEPTION REPRESENTATION The main objective of the Semantic Perception Layer (SPL) is to provide information about in-

21. **fma:**http://www.obofoundry.org/ontology/fma.html (accessed on 26/04/2017)
22. **doid:**http://www.obofoundry.org/ontology/doid.html (accessed on 26/04/2017)

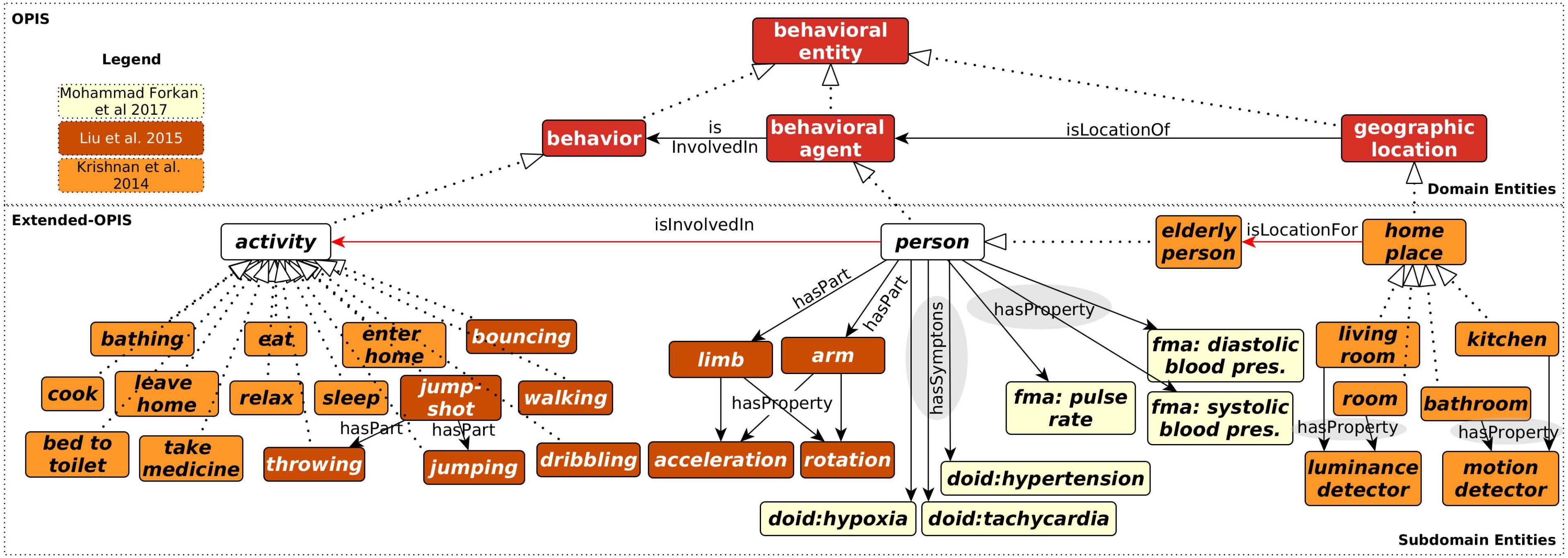

Figure 5.13 – An example of personal information representation from three use case. Colors represent the use case and are described in the legend. Activity and person are not colored because it is used in more than one use cases. Red line arrows represent *behavioral entity properties* that exist due to the restrictions declared between super-classes. Dotted line arrow represent subsumption relationship.

formation processing scenarios that were not capable of being represented using the SSN-O. Figure 5.14 illustrates this information in class assertions (instances) for the OPIS concepts in the specification, execution, and implementation levels, highlighting the information flow, its origin, and result. The ViSiBiD approach [206] is implemented using the *Random Forest* classification algorithm from Weka toolkit and has a window size parameter of 30. Other data processing and data mining techniques, such as segmentation feature computations, used to extract features in this use case, are not described in the illustration for the matter of readability.

In the specification level, the scenario has objective specification of the SPP and algorithms that define the SPP input (rotation property #1 and acceleration property #1) and output (disease#1). Annotations in these class assertions can describe in a human-readable format relevant information, such as ViSiBiD objective description *disease detection*.

In the execution level, the same workflow is represented using the virtual sensor execution class assertion, along with the parts of its execution input (virtual sensor input #1, #2) and result (virtual sensor output #1). Each of data examples (#1 to #3, #4 to #6) has an ssn:observationValue of a gyroscope (related to the rotation ssn:property described in Figure 5.13), and accelerometer (associated with the acceleration ssn:property). The output (virtual sensor output #1) has a data example (data example #7) with a defined precision (precision data example specification #1 hasValue 95.18f), corresponding to a semantic observation that contains a class axiom (fma:hypertension) and a class assertion (hypertension #01).

Lastly, the implementation level contains the specification of a window size parameter that is used to aggregate sensor data during the data processing. In addition, it provides the information about the *Random Forest* implementation, such as an identifier (weka.classifiers.trees.RandomForest) and an IRI.

5.9.2 Use Case: an illustrative scenario for activity perception

The variety of possible use cases using OPIS is as wide as those observed in the SSN-O. We focus on presenting the TBox reasoning capacity related to the provenance provided by OPIS, which permits gathering information about the Semantic Perception Process (SPP) and the implementation parameters. The objective is to show how this process can be described using OPIS and how it represent data quality, data provenance, and information retrieval through examples of SPARQL queries, answering the competency questions presented in Table 5.1.

For this use case, we consider the process of inferring human activities from geographic location points, as described in [207] and presented in Algorithm 1. In this case, human activities are personal information classified as extensions of the *behavior* concept. The algorithm has a set of geographic location points (trajectory T) as input

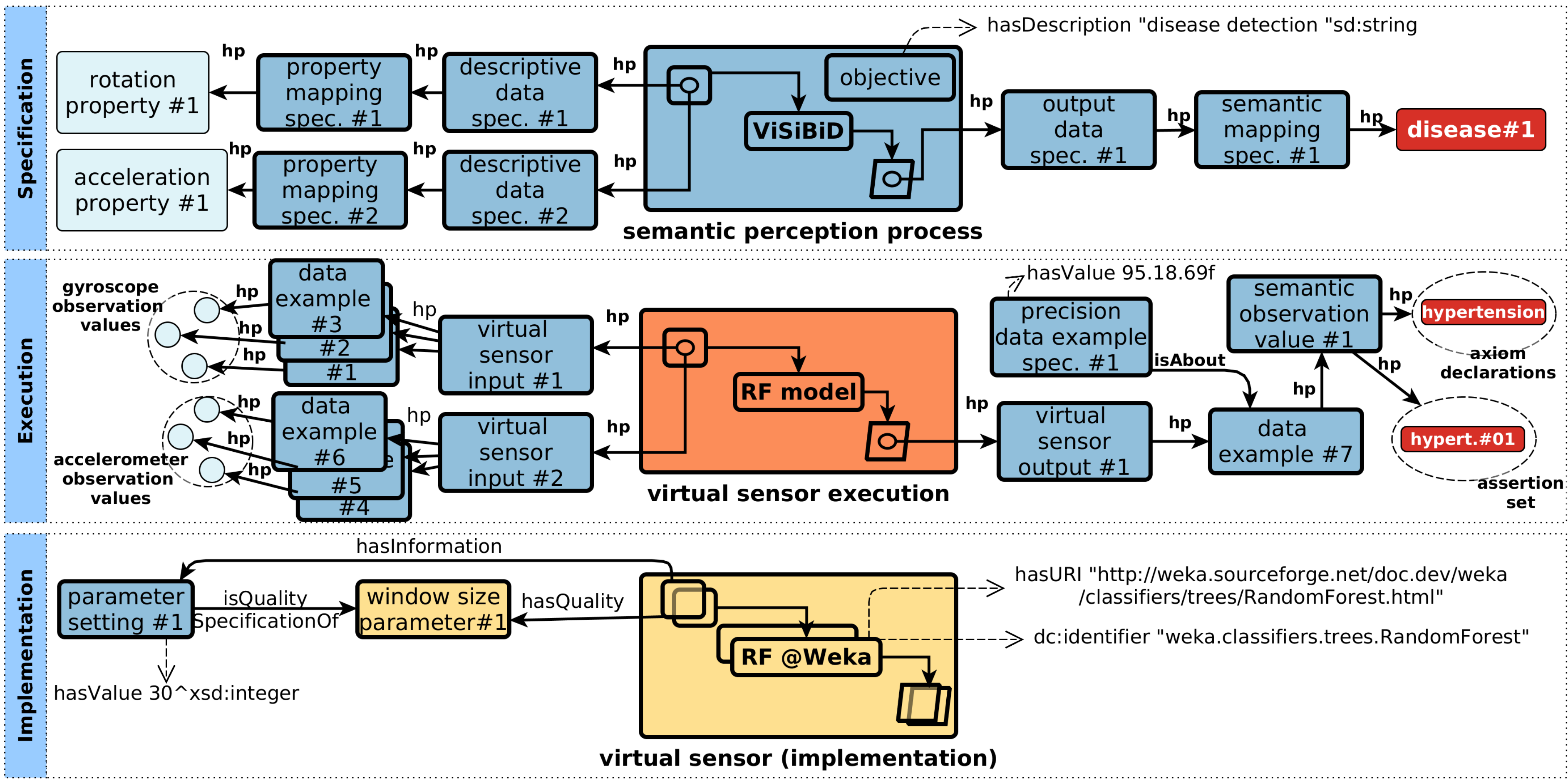

Figure 5.14 – An example annotation for human activity perception (a fragment)

Input : A trajectory T
Output : The activities done during the stops

Stops ← `StopDetection` (T, $\delta_{spatialAccuracy}$, $\tau_{temporalAccuracy}$);
foreach *stop in Stops* **do**
    PossiblePOIs ← `SelectedPOIs` ($stop_{point}$, $stop_{time}$, $WalkingDistance_{max}$);
    Activity ← `Probability` (PossiblePOIs);
    return Activity;
end

Algorithmus 1 : Algorithm for inferring human activities from geographic location points [207].

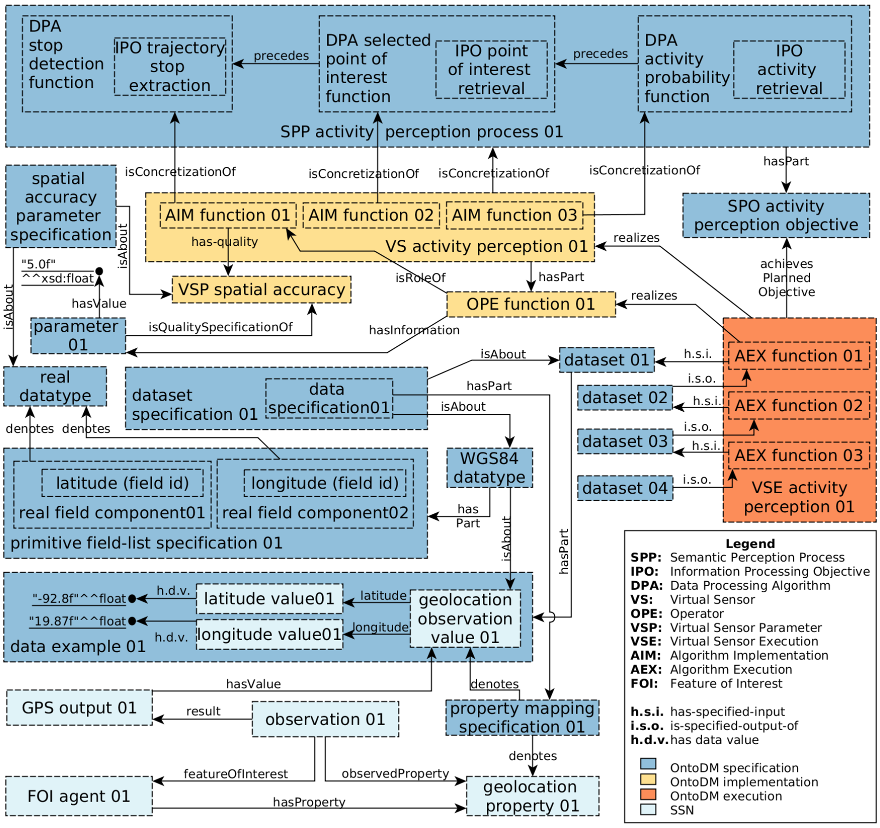

Figure 5.15 – An example annotation for human activity percetion (a fragment)

and produces the descriptions of activities done during the stops. It starts with function StopDetection, calculating stops according to spatial and temporal thresholds. Then, for each stop, the algorithm retrieves possible points of interest (SelectedPOIs), according to a maximum distance. Finally, the activity is computed based on the probable points of interests visited by the agent (Probability).

Figure 5.15 presents a fragment of the representation of Algorithm 1 using OPIS, simplified to be illustrated. The *semantic perception process* (SPP) (**activity perception process 01**) has a *semantic perception objective* (SPO) (**activity perception objective**) that may be annotated with description, name and author data properties. The **SPP activity perception process 01** comprises three algorithm specification related to the functions – *obi:data processing algorithm* (DPA): **stop detection function**, **selected point of interest function**, and **activity probability function**.

The OntoDM object property of *odm:precedes* allows describing the order that each algorithm has in the process execution. In addition, the *odm:information process objective* (IPO) is part of each algorithm specification, which, similarly to the **SPO activity perception objective**, may have specified description, title, author, or any other supplementary information that can be used to analyze a *virtual sensor* based on its process and objective.

For the matter of presentation, the relations between *information processing objectives* and *data specifications* are not illustrated in Figure 5.15. Even so, the **data specification 01** denotes the data type specification (**WGS84 datatype**), being part of the **IPO trajectory stop extract**. In turn, the **property mapping specification 01** that denotes the **geographic location property** is also part of the **data specification 01**, making possible to infer which properties are related to each algorithm. Mapping specification is explained in more detail later.

The *virtual sensor* (VS) **activity perception 01** concretizes the **activity perception process 01** and is composed of three *odm:algorithmImplementations* (AIM) (**AIM function 01,02,03**), that concretizes their respective algorithm specification as defined in the **SPP activity perception process 01**. Each *algorithm implementation* plays a role of an *odm:operator* (OPE) that holds information about algorithm implementation *odm:parameters* (VSP). In particular, the **AIM function 01** that implements the **IPO stop detection function** has a **spatial accuracy** – virtual sensor parameter– to tune the computation of trajectory stops, referenced as $\delta_{spatialAccuracy}$ in Algorithm 1. This *parameter* is specified according to (*iao:is about*) a **spatial accuracy parameter specification** which contains information about parameter name and description, denoting a **real datatype**. In Figure 5.15, only the odm:operator related to **AIM function 01** is depicted for the matter of presentation clarity.

The **OPE function 01** sets the information specified by the **spatial accuracy**. This semantic annotation allows retrieving the conditions (*odm:parameterSettings*) on which *personal information* (*behavior entity*) is generated, which, in this example, is directly associated with the data quality of the inferred human activity.

The information about *semantic process specification* and *virtual sensor* permits reasoning about similarities among *virtual sensors*. In a situation where *virtual sensors* are verified before its execution, for example, anticipating *virtual sensor*'s output or misbehavior, this can support IoT platforms to prevent virtual sensor execution.

The hierarchical structure found both in the specification and in the implementation is equally followed in the execution, where each algorithm execution (including the *virtual sensor execution* itself) achieves a planned objective. At the execution level, each function has specified inputs and outputs. Each dataset has (*ro:has part*) *iao:dataExample* (**data example 01**), which are composed of (*ro:has part*) *ssn:observationValues*. As defined in SSN, ssn:observationValues are annotated with specific unit of measure. In this use case, we are considering WGS84 standard for the geographic location values.

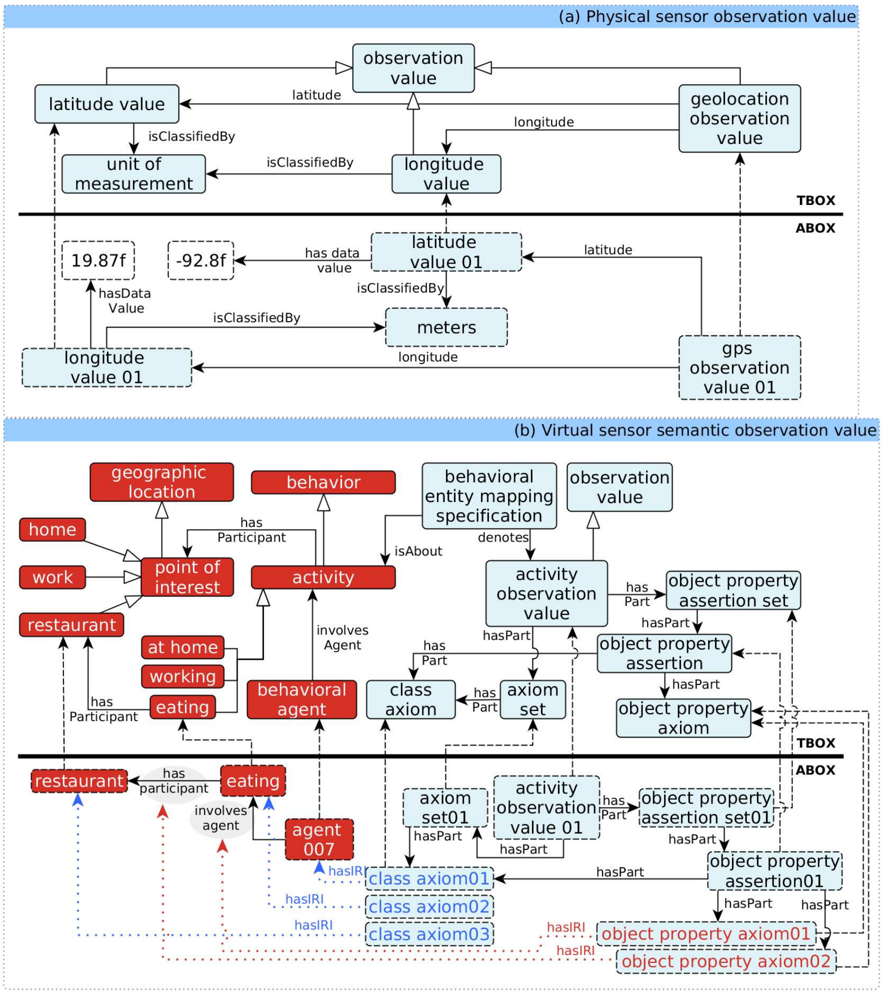

Figure 5.16 – An example of TBox and ABox instances of datatypes, observation values and related behavioral entities. Dash-dotted arrows represent instance-of relationships (rdf:type). Dotted arrows represent hasIRI data annotations, which red are associated with *object property axioms* and blue are associated with *class axioms*.

It is important to note the subtle relationship between the OntoDM specification level of information processing entities, such as algorithms and data mining scenarios, and the representation of the physical world described in SSN-O through sensors and features of interest. The pre and post-*virtual sensor* execution representations provide a two level mapping to conceptual and observational mapping. In OPIS, the mapping specification is extended to represent this mapping at two levels: i) the specification level, denoting SSN-O *properties* (**geographic location property 01**); and ii) the execution level, using *data example* (**data example 01**) as a set of ssn:observationValues (**geographic location observation value 01**).

Regarding the content of *virtual sensor input* and *virtual sensor output*, the *data example* concretely comprises the parts of ssn:observationValues (from SSN-O sensors) and *semantic observation values* (from *virtual sensors*) according to a specific *odm:datatype*. Figure 5.16 presents the input and output specification for the Algorithm 1. Despite the fact that the original algorithm output is a description of human activity, it is possible to represent and associate semantically it using OPIS. A conceptual set based on *behavioral entities* can be used to model the personal information. The inference about human activity, based on geographic location points, is related to the classification of points of interests, such as restaurants and workplace, as described in [207]. In Figure 5.16.a, the specialization of ssn:observationValue in geographic location observation value contemplates the WGS84 data type structured (see Figure 5.12). In Figure 5.16.b, the set of *behavioral entities* and object properties used to represent the result of the Algorithm 1 has activity (*behavior* specialization), point of interest (*geographic location* specialization) and *behavioral agent*.

Therefore, the instances of sensing data and *personal information* are concretely representing the result from a physical sensor (as showed in Figure 5.16.a), and from a *virtual sensor execution* (as illustrated in Figure 5.16.b). Hence, the set of semantic annotation provided by OPIS allows relating physical sensor data to *personal information* (*behavioral entities*). As showed in Figure 5.15, geographic location points can be associated with any *behavioral entity* through object property paths defined, using the set of object properties defined on PIL. For each OPIS class assertion (individual), an instance of the *class axiom* is generated and related to it through the annotation property *hasIRI*. Similarly, object properties that assert a relationship between two individuals is associated with an instance of *object property axiom* through the annotation property *hasIRI*. In the figure, these are represented respectively by the group of **class axiom 01**,**class axiom 02**,**class axiom 03** and **object property axiom 01** and **object property axiom 02**.

The **dataset specification 01** describes (*iao:is about*) *virtual sensor input* and *virtual sensors outputs*, containing **data specification 01** which relates **property mapping specification 01** and the **WGS84 datatype**. The **property mapping specification 01** concretely links **geographic location properties 01** to the iao:dataSpecification, and then, to each part of the **data example 01**.

Finally, it is important to note that the quality associated with a perceived semantic is related to the type of algorithm used to infer the information and the parameter settings. In this use case, the spatial threshold to calculate points of interest can influence directly in the confidence level of the result.

5.9.3 Competence Questions

The capacity to express natural language questions using semantic representations and queries is a manner to demonstrate the ontology competence. Table 5.10 presents semantic representation for the questions proposed in the begin of this chapter.

Q_n	SPARQL Query
Q1	Type(_:sobsv, opis:semanticObservationValue). Type(_:obsv, ssn:observationValue). Type(_:dtExample1, odm:dataExample). Type(_:dtExample2, odm:dataExample). Type(_:vsinput, opis:virtualSensorInput). Type(_:vsoutput, opis:virtualSensorOutput). Type(_:vsexec, opis:virtualSensorExecution). Type(_:classAxiom, opis:classAxiom). PropertyValue(_:dtExample1, ro:hasPart, ?obsv). PropertyValue(_:vsinput, ro:hasPart, ?obsv). PropertyValue(_:vsexec, obi:hasSpecifiedInput, _:vsinput). PropertyValue(_:vsexec, obi:hasSpecifiedOutput, _:vsoutput). PropertyValue(_:vsoutput, ro:hasPart, _:sobsv). PropertyValue(_:dtExample2, ro:hasPart, _:sobsv). PropertyValue(_:sobsv, ro:hasPart, _:classAxiom). Annotation(_:classAxiom, hasIRI, ?be). SubClassOf(?be, behavioralEntity). FILTER (?obsv,:X)
Q2	Type(?vs, opis:virtualSensor). Type(_:spp, opis:semanticPerceptionProcess). Type(_:sems, opis:semanticMappingSpecification). Type(?be, opis:behavioralEntity). PropertyValue(?vs, obi;isConcretizationOf, _:spp). PropertyValue(_:spp, ro:hasPart, _:sems). PropertyValue(_:sems, iao:isAbout, ?be). FILTER (?be,:X)
Q3	Type(?behavior, opis:behavior). Type(?be, opis:behavioralEntity). PropertyValue(?be, dul:isParticipantIn, ?behavior). FILTER (?be=:X)
Q4	Type(?spp, opis:semanticPerceptionProcess). Type(?vs, opis:virtualSensor). Type(_:ope, odm:operator). Type(_:algImp, odm:algorithmImplementation). Type(?setting. odm:parameterSetting). Type(?quality, odm:parameter). PropertyValue(?vs, obi:isConcretizationOf, ?spp). PropertyValue(?vs, ro:hasPart, _:ope). PropertyValue(_:ope, ro:roleOf, _:algImp). PropertyValue(_:algImp, obi:hasQuality, ?quality). PropertyValue(_:ope, exact:hasInformation, ?setting). PropertyValue(?setting, iao:isQualitySpecificationOf, ?quality). FILTER (?spp=:X)
Q5	Type(?be, opis:behavioralEntity). Type(_:spp, opis:semanticPerceptionProcess). Type(_:smes, opis:semanticMappingSpecification). Type(?algorithm, obi:algorithm). Type(?objective, iao:objectSpecification). PropertyValue(_:spp, ro:hasPart, _:sems). PropertyValue(_:spp, ro:hasPart, ?algorithm). PropertyValue(?algorith, ro:hasPart, ?objective). FILTER (?be=:X)
Q6	Type(?vs, opis:virtualSensor). Type(_:spp, opis:semanticPerceptionProcess). Type(_:smes, opis:semanticMappingSpecification). Type(?algorithm, obi:algorithm). Type(?objective, iao:objectSpecification). PropertyValue(?vs, obi:isConcretizationOf, _:spp). PropertyValue(_:spp, ro:hasPart, ?algorithm). PropertyValue(?algorith, ro:hasPart, ?objective). FILTER (?vs=:X)
Q7	Type(?machine, obi:computer). Type(?be, opis:behavioralEntity). Type(?vsExec, opis:virtualSensorExecution). Type(_:vsoutput, opis:virtualSensorOutput). Type(_:classAxiom, opis:classAxiom). Type(?algorithm, obi:algorithm). PropertyValue(?vsExec, ro:hasAgent, ?machine). PropertyValue(?vsExec, obi:hasSpecifiedOutput, _vsoutput). PropertyValue(_:vsoutput, ro:hasPart, _:classAxiom). Annotation(_:classAxiom, opis:hasIRI, ?be) FILTER (?be=:X)
Q8	Type(?spp, opis:semanticPerceptionProcess). Type(_:pms, opis:propertyMappingSpecification). Type(?property, ssn:property). PropertyValue(?spp, ro:hasPart, _:pms). PropertyValue(_:pms, iao:isAbout, ?property). FILTER (?spp=:X)

Table 5.10 – Formalization of competency questions using the SPARQL-DL notation.

5.10 CONCLUSION

We presented OPIS, an ontology for personal information on the Sensor Web. OPIS is competent to describe *personal information* by leveraging the semantics of the sensor network to describe Semantic Perceptions (SPs) that infer personal information. We employed the concepts from the BC to model the set of information that could be considered as personal. As consequence, a set of classes (*behavioral entities*) and properties are proposed to represent personal information. Moreover, the descriptive and realist perspectives used in our approach provide a cognitive interface for individuals to better understand and classify their information, in the same time that offers a realist viewpoint normally adopted to describe the physical, mental, neural, biological, and environmental world. Along with this, we represent KDDM processes using the concept of virtual sensors, encapsulating the process of personal information inference while associating to it a semantic annotation.

6 PA-VSM: PRIVACY-AWARE VIRTUAL SENSOR MODEL

CONTENTS

6.1 INTRODUCTION

In this chapter, we introduce our privacy mechanism for the IoT sensing based on a *privacy-by-policy* strategy and the compositional and modular Privacy-aware Virtual Sensor Model (PA-VSM) as *privacy-by-design*. Our approach adopts the principle of *plurality* of Privacy-Preserving Data Mining Techniques (PPDMTs), applying selectively these techniques according to *contextuality* and *contestability* defined in the privacy policy. The *contextuality* is based on the expressiveness of OPIS to describe personal information according to the behavioral context, while the *contestability* relies on the capacity to represent KDDM and reason about inference intentions of Semantic Perception Process (SPP). In addition, the PA-VSM introduces *privacy-by-design* implementing a *joint sphere* paradigm, shifting the privacy mechanism from the *recipient sphere* to the cloud.

The advantage of incorporating Semantic Web technologies to our solution is the capacity of reasoning over OWL Description Logic (OWLDL) [208], using any *off-the-shelf* DL reasoner to process Ontology Web Language (OWL) axioms using SPARQL. The formality of OWL allows an unambiguous meaning representation of personal information type, context, SPPs, PPDMTs, and policy conditions, making possible to draw conclusions about inference intentions that are important to *privacy-by-policy* mechanisms.

The remainder of this chapter is organized as follows. Section 6.2 we present the ontological framework to provide an extensive classification scheme for personal information. In Section 6.3, we present our

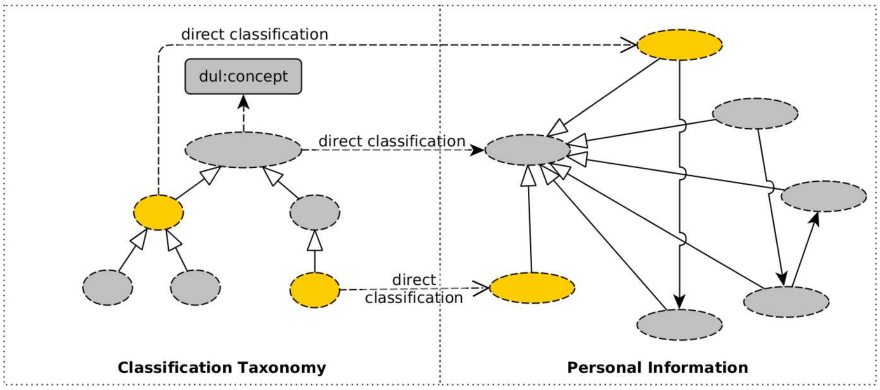

Figure 6.1 – Classification Taxonomy vs Personal Information direct mapping schema.

novel privacy-aware virtual sensor model for the IoT sensing. Lastly, Section 6.4 presents conclusions.

6.2 ONTOLOGY-BASED PRIVACY-BY-POLICY

6.2.1 Personal Information Classification

The classification of personal information using OPIS can admit several possibilities based on classifying behavioral entities directly or through its taxonomy-like structure, assuming that subclasses are classified according to its superclasses. However, these approaches restrict the re-usability of classification structures and couple classification to the ontology axioms. One way to address this problem is providing an external classification structure to index concepts that classify entities using mapping functions. We propose an external a taxonomy as classification structure. This approach provides a classification structure targeting the same set of axioms (OPIS), which is reusable, decoupled and flexible.

DIRECT CLASSIFICATION The basic case consists of a *direct classification* between the *classifier* and a *classified entity*, where the classifier is defined in a classification taxonomy, as depicted in Figure 6.1. The taxonomy root classification is the most top classifier and considered to be the classifier by *presupposition* [1]. In this thesis, we employ the descriptive ontological framework of DUL to specify concepts (*dul:concept*) to be used as a classification taxonomy. The dul:concept can be used to classify OPIS axioms (class axiom, object property axiom, or data property axiom) directly through the object property *dul:classifies*. As described in chapter 5, OPIS axioms, or any extension of them, represent personal information. In Figure 6.2 we depict two examples of this classification taxonomy. One illustrates concepts of aspects of human life, such as social, professional, and homelike, that may be used to classify OPIS axioms. The other depicts the specializa-

1. Assumption that all elements have this property if nothing else is asserted.

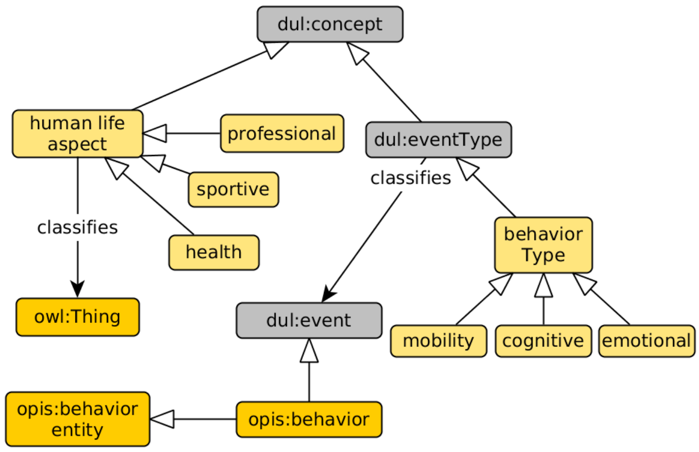

Figure 6.2 – Example of classification based on behavior types and human life aspects.

tion of dul:eventType to represent behavior types, such as mobility, cognitive, or emotional.

In order to define formally the concepts in this chapter, we need to provide formalization for *RDF term* and *RDF graph*, which are the base to represent and exchange information that is retrievable using SPARQL queries. We define formally *RDF terms* and *RDF graphs* according to the definitions described in [209] as follows:

Definition 1 (RDF Term) *Let **i**, **l**, **b** be an IRI, **rdf:literal**, and **blank nodes** respectively. Denote **I**, **L**, and **B** as infinite disjoint sets of IRIs, rdf:literals, and blank nodes respectively, then:*

— *the set of all RDF terms **T** is $\mathbf{I} \cup \mathbf{L} \cup \mathbf{B}$;*
— *a **RDF triple** is a triple (s,p,o) from $\mathbf{T} \times \mathbf{I} \times \mathbf{T}$;*
— *a **RDF graph** is a finite set of RDF triples.*

Most of the *syntactic forms* of OWLDL have equivalent *syntactic sugar* forms, which are common axiom constructors or statements used to make ontology construction and maintenance easier. For example, the *disjointWith* statement can be used to declare that class X *disjointWith* class Y. This is a *syntactic sugar* for the axiom $X \sqsubseteq \neg Y$. Aiming to simplify the definition in this chapter, we adopted the SPARQL-DL Language (SPARQL-DL) query atoms, as defined in [210], referring to it using the namespace *owldl*. The foundation for SPARQL queries is the *basic graph patterns*, which are formed by *triple patterns*. Both concepts are defined formally as follows:

Definition 2 (Triple and graph patterns) *Denote **V** as an infinite set {?x, ?y, ...} of query variables disjoint from **T** where:*

1. *a **triple pattern** is an element $\boldsymbol{tp} \in (\mathbf{T} \cup \mathbf{V}) \times (\mathbf{I} \cup \mathbf{V}) \times (\mathbf{T} \cup \mathbf{V})$;*
2. *a **basic graph pattern** is a set of triple patterns.*

As defined in Section 3.1.3, *property paths* can be specified based on RDF terms and used to query RDF graphs concisely. *Property paths* simplify triple patterns, which are the basic unit to describe RDF graph using RDF triple patterns. Syntactically, *property paths* are expressions defined according to a specific grammar. The *property path expression* that defines the syntax of *property paths* are formalized following the definitions in [108], as described below:

Definition 3 (Property path expression) *Property path expression is defined recursively according to the grammar:*

$$e = i|e^{-}|e_1 \cdot e_2|e_1 + e_2|e^{+}|e^{*}|e?|!\{i_1 \ldots i_n\}|!\{i_1^{-} \ldots i_n^{-}\},$$

where $\boldsymbol{i}, \boldsymbol{i}_1, \ldots \boldsymbol{i}_n \in \boldsymbol{I}$, and the operators have the same semantics defined in Table 3.2: $\boldsymbol{e}^{-}$ represents an inverse path ^$\boldsymbol{e}$, $\boldsymbol{e}_1 \cdot \boldsymbol{e}_2$ represents the sequence operator (/), and $\boldsymbol{e}_1 + \boldsymbol{e}_2$ refers to alternative paths (|). The last two forms correspond to negated property sets.

Denote **E** the set of all *property path expressions*. *Property path expressions* are therefore incorporated on the atomic level of SPARQL by means of triples with IRIs (**I**), *rdf:literals* (**L**) and variables (**V**) on the edges and property path expressions (**E**) between them.

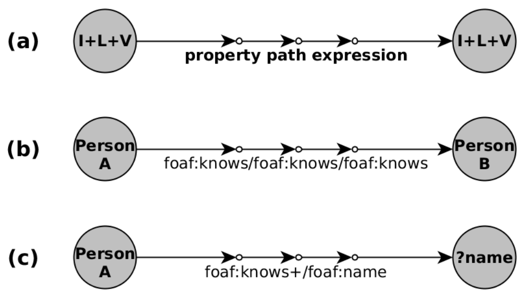

Figure 6.3 – (a) Property path pattern representation. (b) Property path example using FOAF. (c) Property path pattern example using FOAF

This pattern is depicted in Figure 6.3.(a), along with two examples of *property paths* and *property path pattern*. *Property path patterns* are defined formally as follows:

Definition 4 (Property path pattern) *A property path pattern is a triple in ($\boldsymbol{I}$∪$\boldsymbol{L}$∪$\boldsymbol{V}$)×$\boldsymbol{E}$×($\boldsymbol{I}$∪$\boldsymbol{L}$∪$\boldsymbol{V}$)*

By using the SPARQL-DL query language to define an RDF graph, we assume that exists at least one valid *evaluation mapping* in the RDF graph that satisfies the query. An *evaluation mapping* is a result of an evaluation function $[\![P]\!]_{\mathcal{G}}$, where P is a *basic graph pattern* over the RDF graph $\mathcal{G}$, as formally defined in [211].

For example, by defining the RDF graph $\mathcal{G}$ using the query atom "SubClassOf(A,B)", we assume that there is an *evaluation mapping* for the RDF graph form "A owldl:SubClassOf B" of this SPARQL-DL query atom in $\mathcal{G}$. Thus, if the RDF triple "A rdfs:subClassOf B" or the RDF graph "A rdfs:subClassOf [rdfs:subClassOf B]" exists in $\mathcal{G}$, they are valid *evaluation mappings* for "A owldl:SubClassOf B", and $\mathcal{G}$ satisfies the SPARQL-DL query "SubClassOf(A,B)".

Based on these concepts, the *classification taxonomy* are RDF graph formed by RDF triples that specify subclasses of dul:concept and annotation properties related to them. We formally define *classification taxonomy* as follows:

Definition 5 (Classification Taxonomy) *The classification taxonomy is an RDF graph $\mathcal{C}$ formed by the union of two RDF graph $\mathcal{O}$ and $\mathcal{A}_{\mathcal{O}}$, iff:*
— $[\![$ *SubClassOf(c, dul:concept)* $]\!]_{\mathcal{O}}$ *is satisfied;*
— $[\![$ *Annotation(c , a , dul:concept)* $]\!]_{\mathcal{A}_{\mathcal{O}}}$ *is satisfied and* $c \in \mathcal{O}$.

Nonetheless, it is important to remark that we develop in this chapter the notions of personal information classification from a descriptive perspective, i.e. using DUL classes. Although, the mapping between DOLCE-based and BFO-based entities, presented in the previous chapter, allows applying the realist perspective to represent and classify behavioral entities, using the same definitions and mechanisms described in this chapter.

A *direct classification* is an RDF graph constituted by RDF triples, representing class assertions that contain one annotation property (hasIRI) that points to personal information – class axiom or object property axiom –, class assertions of dul:concepts (or subclasses of them) and object property assertions relating both class assertions. Based on these concepts, *direct classification* can be formally defined as follows:

Definition 6 (Direct Classification) *Be $\mathcal{O}$ an RDF graph that represents individuals of personal information, $\mathcal{C}$ an RDF graph that represents the classification taxonomy, and **hasIRI** a data property which range is xsd:anyURI. A direct classification is an RDF graph $\mathcal{D}$ formed by the union of four RDF graphs $\mathcal{IC}$, $\mathcal{CO}$, $\mathcal{IT}$, and $\mathcal{IO}$, iff:*
— $[\![$ *Type(c, dul:concept)* $]\!]_{\mathcal{IC}}$ *is satisfied, and* $c \in \mathcal{C}$;
— $[\![$ *Type(o, owl:Thing),Annotation(o, hasIRI, i), SubClassOf(i, owl:class)* $]\!]_{\mathcal{IL}}$ *is satisfied,* $o \in \mathcal{O}$ *and* $i \in I$;
— $[\![$ *Type(o, owl:Thing),Annotation(o, hasIRI, i), SubPropertyOf(i, owl:ObjectProperty)* $]\!]_{\mathcal{IO}}$ *is satisfied,* $o \in \mathcal{O}$ *and* $i \in I$;
— $[\![$ *PropertyValue(c, dul:classifies, o)* $]\!]_{\mathcal{CO}}$ *is satisfied;*

TRANSVERSAL CLASSIFICATION Aiming to take advantage of the semantic relationships among classes, we extend the concept of *direct classification* to contemplate extensional and intensional abstraction levels. In the extensional abstraction level, classification is transitively transferred from a class to its subclasses, while in the intensional abstraction level classification is transitive from a class to another through object properties and axiom expressions.

The conceptual layers of OPIS provide the foundation to describe personal information using behavioral entities, as defined in Table 5.5. In fact, OPIS classes are designed to be extended in sub-domain concepts, similarly to the usage of the Semantic Sensor Network Ontology (SSN-O). For example, the class *behavioral agent* can be extended to represent a patient, doctor, or runner. This rationale is analogous to the object properties, that can be extended to specialize their intension. For instance, the object property *isAcknowledgedBy* can be specialized as isLearnedBy or isConsideredBy. The intentional abstraction level is constituted by the object properties that relates one entity to another, as initially defined in Table 5.6. Moreover, in PIL behavioral entity represents three abstraction levels: i) behavioral entity classes and behavioral entity properties (Vectors 5.1 and 5.2); ii) behavior; and iii) behavior pattern. Since we define *behavioral agent* as central "hubs" that needs to be related to other behavioral features on classes and object property extensions, we guarantee that an intensional abstraction level is also defined along with the extensional abstraction level.

Based on this ontological framework, we define the concept of *classification transitivity* that allows classifying a graph of connected entities transversally based on their sets of object properties and subclasses. While the *direct classification* was represented by an RDF graph, the concept of *classification transitivity* is based on the results of SPARQL queries due to its capacity to represent *property paths*, and therefore, *transversal classifications*.

In order to restrict *property paths* to express only *transversal classification* property path expressions, we need to describe the expression structure of *property paths* and *property path patterns* firstly. The intended *transversal classification* is achieved by defining the *property path pattern* that represents the two previously mentioned abstraction level of *property paths*: *intensional property path* and *extensional property path*.

Transversal classification is illustrated in Figure 6.4 using the example of the use case described in Section 5.9.2. We separate the base OPIS entities from the extended set that was used to represent information resulted from Algorithm 1. As defined originally in OPIS, *behavior* has participants, among them *behavioral agents* and *geographic location*. *Behavior* is specialized to represent activities of the type eating, working, or atHome. *Geographic location* is extended to represent points of interest of the type work, home, and restaurants. In addition, we increment the example by extending the class *behavioral agent* as *engineers* and adding an object property *dul:isLocationOf* to represent the intensional relation between some activities and points of interest.

The *direct classification* associates the *family context* (dul:concept) to the activity of type *atHome*. By using SPARQL queries, it is possible to retrieve all involved behavioral entities at the same level of abstraction that may have transitively the same classification. For example, by querying all classes that are related to *atHome* activity based on the *isLocationOf* object property, it is possible to retrieve the point of interest *home*, and consequently, classifying it transversally.

Besides that, Figure 6.4 illustrates different abstraction levels, depicting how classification can be hierarchically transitive by extension. Classifications are denoted by colors: yellow for *private* concept, and gray for *non-private* concept. The *transitive classification* is denoted by dash-dotted arrows, representing an *extensional property path* (engineer subClassOf behavioralAgent) and an *intensional property path*

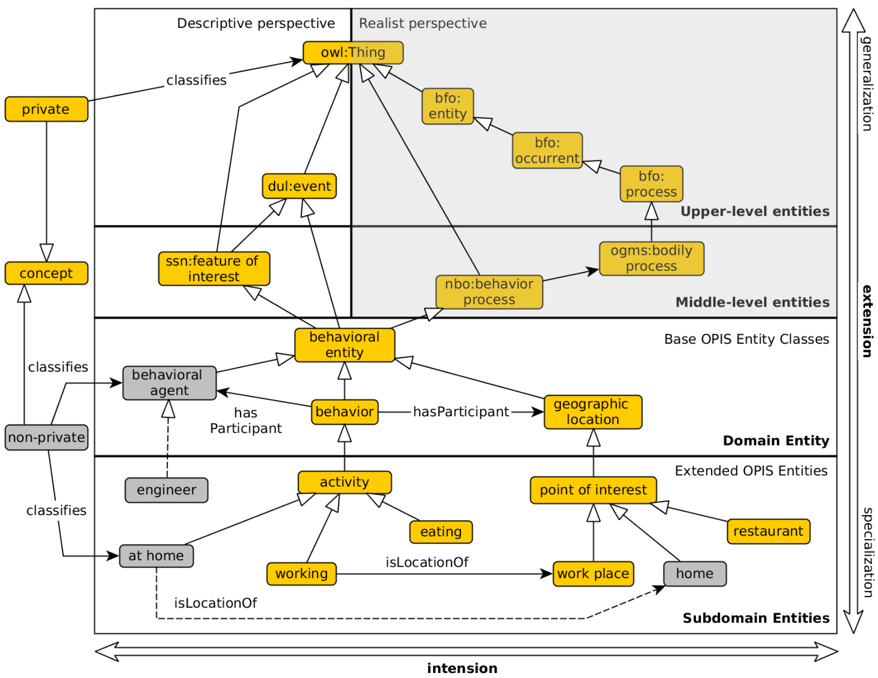

Figure 6.4 – An example of horizontal and vertical classification. Yellow ellipses represent the semantic entities classified according to private classification. Gray ellipses represent entities classified according to non-private classification. isf is abbreviation for isSettingFor.

(atHome *happensInSpecificPointOfInterest* home). In addition, Figure 6.4 depicts the mapping of descriptive and realist perspectives that may be used during the personal information classification.

The *extensional property paths* correspond to *property paths* that contain only RDFS constructors that define specialization/generalization, such as *rdfs:subClassOf* and *rdfs:subPropertyOf* (see Section 3.1.2). The *intensional property paths* consist of *property paths* that refer to any other IRIs that do not comprise the specialization/generalization RDFS constructors, enclosing *property paths* to the same level of abstraction. We do so by restricting *property path expressions* to be formed only with a subset of IRIs (**I**). We also assume that cyclic paths can happen but enclosed in these two dimensions, i. e. according to the following definitions, cyclic paths will be based only in RDFS constructors or completely without them.

We formally define *intensional property path expression* and *extensional property path expression* as follows:

Definition 7 (Intensional and extensional property path expressions) *Denote* $\boldsymbol{I}_v$ *= {owl:subClassOf, owl:subPropertyOf}, and* $\boldsymbol{I}_h = \boldsymbol{I} - \boldsymbol{I}_v$.

— *The* ***intensional property path expression*** *is defined by the grammar*

$$h = \boldsymbol{i} | h^{-} | h_1 \cdot h_2 | h_1 + h_2 | h^{+} | h^{*} | h? | !\{i_1 \dots i_n\} | !\{i_1^{-} \dots i_n^{-}\},$$

where $\boldsymbol{i}, \boldsymbol{i}_1, \dots \boldsymbol{i}_n \in \boldsymbol{I}_h$*;*

— *The* ***extensional property path expression*** *is defined by the grammar*

$$s = \boldsymbol{i}_v | s_1 \cdot s_2 | s_1 + s_2 | s^+ | s^* | s? | !s,$$

where $\boldsymbol{i}_v \in \boldsymbol{I}_v$.

Regarding the *extensional property path pattern*, it is evident that *classification transitivity* is only valid from superclasses toward subclasses. In the example of Figure 6.4, it does not make sense, for instance, to assume that *activity* is classified as *non-private* because of its relation of super-class with *atHome* entity. Therefore, we define *extensional property path expressions* formally by restricting its grammar to be based only on *owl:subClassOf* and *owl:subPropertyOf*, avoiding negative forms that could change the intended direction of the property path.

When dealing with *transversal classification*, it is necessary to restrict the expression of *property path* in order to guarantee the transversal order of path concatenation as described above. *Transversal classification* can be transitive only towards the class extension direction or at the same level of intension abstraction through object properties. The *transversal classification property path*, therefore, can be expressed using a regular expression based on intensional and extensional property path expressions, being formally defined as follows:

Definition 8 (Transversal classification path expression) *Let* $\boldsymbol{v}$ *be an extensional property path expression, and* $\boldsymbol{h}$ *is an intensional property path expression. A* ***transversal classification path expression*** *is defined as*

$$\boldsymbol{c} = \boldsymbol{h} \mid \boldsymbol{v} \mid c_1 \cdot c_2 \mid c_1 + c_2,$$

where $c_1 \cdot c_2$ *represents the sequence operator (/) and* $c_1 + c_2$ *refers to alternative paths (|).*

Based on this definition, we define **transversal classification path pattern** to be used in SPARQL queries and to express *transversal classification*, and are formally defined as follows:

Definition 9 (Transversal classification path pattern) *Denote* $\boldsymbol{E}_c$ *as a set of all transversal classification path expressions* $\boldsymbol{c}$. *A transversal classification path pattern* $\boldsymbol{cp}$ *is a triple in* $\boldsymbol{V} \times \boldsymbol{E}_c \times \boldsymbol{V}$;

There are infinite possibilities to define a *transversal classification path pattern*. The *transversal classification path expression* allows recursively many different patterns that satisfy our definition of *classification transitivity*.

As RDF graphs that concretely represent the correspondence between *classifier* and *classified entity*, *direct classifications* can be queried to verifies if a given entity is classified using simple triple pattern, such as:

```
?c dul:classifies ?o,
```

where **?c** and **?o** $\in$ **V** and represent, respectively, classifier and classified entity. A select SPARQL query that considers all variables based on this triple pattern returns a set of IRI records associated to *evaluation mappings* for this triple pattern, representing pairs of *classifier* (dul:concept) and *classified entity* (opis:behavioralEntity).

Conversely, *transversal classifications* are not RDF graphs, but a set of IRI records associated to *evaluation mappings* for a specific *transversal classification path patterns* over an RDF graph $\mathcal{G}$. We then formally define *transversal classification* as follow:

Definition 10 (Transversal classification) *Let $\mathcal{C}$ an RDF graph that represents classifiers (class axioms and class assertions of a classification taxonomy); $\mathcal{D}$ a graph that with one or more **direct classifications**; $\mathcal{O}$ an RDF graph that represents classified class axioms and assertions (which refer to personal information class axiom or object property axiom and their instances, such as those defined in OPIS ontology), and **tc** a transversal classification path pattern. A **transversal classification** is a set of IRI pair of classifier individual and classified individual <ic,il>, associated to one or more **tc**, such that $[\![\; tc \;]\!]_{\mathcal{C} \cup \mathcal{D} \cup \mathcal{O}}$ is satisfied.*

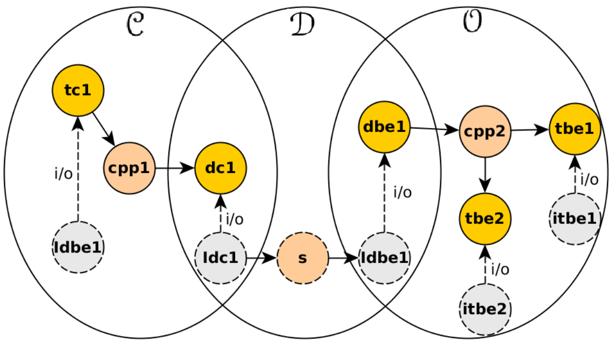

Figure 6.5 – Transversal classification path pattern. Dash-dotted ellipses represent individuals. Dash-dotted arrows represent instance of (i/o).

Figure 6.5 illustrates the union set intersections among the RDF graphs $\mathcal{C}$, $\mathcal{D}$, and $\mathcal{O}$; and the *transversal classification path pattern* that connects non-direct classifiers (tc1) to non-direct classified entities (tbe1,tbe2) through a *direct classification*

```
Idc1 rdf:type dc1. Idc1 s Idbe1. Idbe1 rdf:type dbe1.
```

A RDF graph form of an evaluation mapping for a *transversal classification path pattern* is composed of classification taxonomy property path (cpp1) defined in $\mathcal{C}$, a *direct classification*, and a personal information property path (cpp2) defined in $\mathcal{O}$. In the example, two *transversal classifications* are illustrated:

```
1.idbe1 rdf:type/cpp1/inverseOf(rdf:type)/s/rdf:type/cpp2/rdf:type itbe1,
2.idbe1 rdf:type/cpp1/inverseOf(rdf:type)/s/rdf:type/cpp2/rdf:type itbe2.
```

For the following examples and use cases in this manuscript, we associate the concept of *transversal classification* to the transversal classification path pattern described in Figure 6.5. The conditional part (where clause) of an instance of SPARQL-DL query is presented in Table 6.1 using SPARQL-DL language extension separated in TBox, ABox, RBox for clarity matters. The TBox/ABox type part of the where clause specifies individual assertions. IRIs are specified using the namespace followed by the class axiom, i. e. opis:behavioralEntity, blank nodes are expressed starting by _:, i. e. _:dBE, and variables are represented starting by a question mark ?, such as ?itBE.

clause	type	#id	query
where	RBox	R1	ObjectProperty(AND(_:cpp1,_:cpp2))
	TBox	T1	SubClassOf(_:dBE,OR(opis:behavioralEntity,owl:objectProperty))
		T2	SubClassOf(_:tBE,OR(opis:behavioralEntity,owl:objectProperty))
		T3	PropertyValue(_:dBE,_:ccp1,_:tBE)
		T4	SubClassOf(_:dConcept,dul:concept)
		T5	SubClassOf(_:tConcept,dul:concept)
		T6	PropertyValue(_:tConcept,_:ccp2,_:dConcept)
	TBox/ ABox	TA1	Type(**?itConcept**, _:tConcept)
		TA2	Type(**?itBE**,_:tBE)
		TA3	Type(_:idConcept,_:dConcept)
		TA4	Type(_:idBE, _:dBE)
	ABox	A1	PropertyValue(_:idConcept,dul:classifies,_:iAxiom)
		A2	Annotation(_:iAxiom, opis:hasIRI, _:idBE)

Table 6.1 – An instance of transversal classification path pattern

6.2.2 Privacy-Preserving Virtual Sensor

Our *privacy-by-policy* strategy assesses the following five risk of privacy harm, as described in Section 2.2: 1) data processing techniques; 2) privacy versus public release; 3) data quality; 4) motive; and 5) trust; using the concept of virtual sensors. The first assessment is based on the data processing techniques relies on the *semantic signature of virtual sensors* that allows representing KDDM techniques. The second and third assessments are achieved by the very nature of virtual sensors that shifts the privacy paradigm to a *joint sphere* using the Cloud-IoT infrastructure. The identification of personal information along with the identification of virtual sensors permit to assess the motive of the KDDM technique. Lastly, the trust assessment is based on the concept of certified (trusting) and non-certified (untrusting) virtual sensors.

We extend the concept of virtual sensor to represent the data property of certification. The virtual sensor certification is guaranteed by an independent *trusted party* that provides services to assure and certify that a specific virtual sensor behaves according to its semantic signature. In the remainder of this manuscript, we refer to *trust party* as an organization that provides services to certify and verify certification of virtual sensors.

Besides that, the *plurality* of our *privacy-by-policy* strategy is based on the possibility to employ different Privacy-Preserving Data Mining Techniques (PPDMTs) or Access Control Models (ACMs) according to specific conditions. These PPDMTs, as defined in Section 4, depends

axiom	expression
privacyPreservingObjective	$\sqsubseteq$ iao:objectiveSpecification
privacyPreserving-Technique	$\sqsubseteq$ opis:semanticPerceptionProcess
	$\sqsubseteq$ $\exists$ ro:hasPart privacyPreservingObjective
	$\sqsubseteq$ $\exists$ ro:hasPart (iao:algorithm $\sqcap$ $\exists$ ro:hasPart (iao:objectiveSpecification $\sqcap$ $\exists$ hro:hasPart (odm:dataSpecification $\sqcap\exists$ iao:isAbout odm:datatype $\sqcap$ $\exists$ ro:hasPart opis:propertyMappingSpecification)))
privacyPreserving-VirtualSensor	$\sqsubseteq$opis:virtualSensor
	$\sqsubseteq$ $\exists$obi:isConcretizationOf privacyPreservingTechnique
accessControlObjective	$\sqsubseteq$ privacyPreservingObjective
accessControl	$\sqsubseteq$ privacyPreservingTechnique
	$\sqsubseteq$ $\exists$ ro:hasPart accessControlObjective
	$\sqsubseteq$ $\exists$ ro:hasPart (iao:algorithm $\sqcap$ $\exists$ ro:hasPart (iao:objectiveSpecification $\sqcap$ $\exists$ hro:hasPart (odm:dataSpecification $\sqcap$ $\exists$ iao:isAbout opis:semanticPerceptionDatatype $\sqcap$ $\exists$ ro:hasPart opis:semanticEntityMappingSpecification)))
accessControl-VirtualSensor	$\sqsubseteq$ privacyPreservingVirtualSensor
	$\sqsubseteq$ $\exists$ obi:isConcretizationOf accessControl

Table 6.2 – Privacy-preserving virtual sensor definition

on parameters and specific input datatypes that can be represented using the concept of virtual sensor. Access Control Models (ACMs) are particular cases of privacy-preserving that performs a verification and which results is the unchanged data stream or nothing (if access is denied). In other words, by defining PPDMTs and ACMs as KDDM

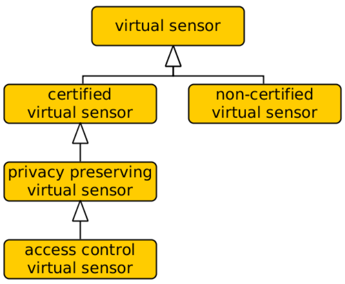

Figure 6.6 – Virtual sensor specialization for privacy-by-policy strategy

processes, which can be represented as virtual sensors, it is possible to specify input and output in order to match data streams to PPDMTs and ACMs in specific conditions. This match is achieved by reasoning over *semantic signature of virtual sensor* based on classes defined in the SPL of OPIS. Figure 6.6 presents the specialization of virtual sensors based on the concepts of *trust* and *plurality*.

In order to represent PPDMT using the concept of virtual sensor, we define privacy-preserving processes and objective specifications, defining Privacy-Preserving Virtual Sensor (PPVS) and Access Control Virtual Sensor (ACVS), as formally described in Table 6.2. The privacyPreservingTechnique is defined as a semantic perception process based on the privacyPreservingObjective and algorithms which have objective specifications constituted by data specifications that are about some datatype and linked to feature of interests and property through property mapping specifications. For accessControl, a subclass of privacyPreservingTechnique, access control objective is defined along with algorithms that have objective specifications constituted by data specifications that are about semantic perception datatype and linked to one or more axioms, such as behavioral entity classes, OPIS object properties or data properties.

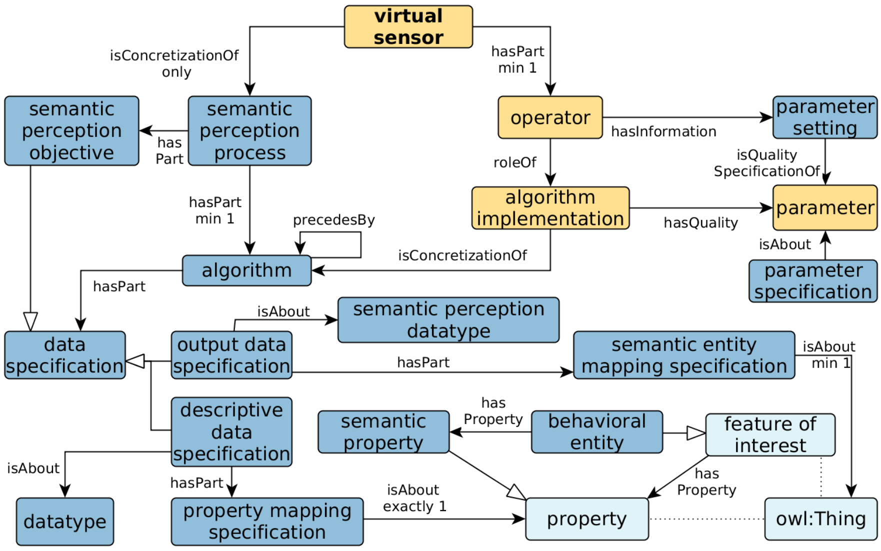

Figure 6.7 – Virtual sensor signature.

SEMANTIC SIGNATURE OPIS ontology can represent three levels of workflow and algorithm: specification, implementation, and execution. Based on these three representation levels, we define the *semantic signature of virtual sensors* based on specification and implementation information. The semantic signature is an RDF graph that contains RDF triples of individuals that can instantiate a specific set of Semantic Perception Layer (SPL) axioms. Figure 6.7 illustrates this set of SPL that should be instantiate as a semantic signature for virtual sensors.

By defining semantic signatures for virtual sensors, it is possible to perform reasoning over signatures using SPARQL-DL queries, retrieving promptly information about virtual sensors needed to implement our Privacy-aware Virtual Sensor Model (PA-VSM). In addition, since these semantic signatures are RDF graphs, similarity calculations between distinct signatures can be used to compare virtual sensors and their inference intentions. Table 6.3 defines each attribute of the semantic signature based on SPARQL-DL queries that should be satisfied in the signature RDF graph. The signature is composed of a virtual sensor identification $\mathcal{CID}$ (class and instance IRIs), a certification $\mathcal{VSC}$, similarity measurements $\mathcal{VSS}$, process specification $\mathcal{VSP}$, algorithms $\mathcal{VSA}$, objective specification $\mathcal{VSG}$, input specification $\mathcal{VSI}$, output specification $\mathcal{VSO}$, and operators $\mathcal{VSE}$.

Similarly, the semantic signature of a Privacy-Preserving Virtual Sensor (PPVS) is specialized to reflect the structure of PPVS. Differently, from virtual sensors, PPVS have the same input and output

attr.	description	SPARQL-DL query
$\mathcal{CID}$	virtual sensor class \ instance id	SubClassOf(class,opis:virtualSensor) Type(id,class)
$\mathcal{VSC}$	certification	Annotation(id,opis:hasCertification,cert)* Type(cert,rdfs:literal)*
$\mathcal{VSS}$	similarity measurement	Annotation(id, ro:hasPart, _:sim)* Type(_:sim, xsd:real)* PropertyValue(_:sim,obi:isSpecifiedOutputOf, _:simCalc)* Type(_:simCalc,obi:similarityCalculation)*
$\mathcal{VSP}$	process specification	PropertyValue(id,obi:isConcretizationOf,_:spp) Type(_:spp,opis:semanticPerceptionProcess)
$\mathcal{VSA}$	algorithms	PropertyValue(_:spp,ro:hasPart,_:algo) Type(_:algo, obi:algorithm) PropertyValue(_:algo,ro:hasPart,_:aos) Type(_:aos,odm:objectSpecification) Annotation(_:aos,_:osa,_:osav) Type(_:osa,owl:AnnotationProperty) Type(_:osav,rdf:Literal)
$\mathcal{VSG}$	objective specification	PropertyValue(_spp, ro:hasPart, _:spos) Type(_:spos,opis:semanticPerceptionObjectiveSpecification) Annotation(_:spos, _:osa, _:osav)
$\mathcal{VSI}$	input specification	PropertyValue(_:aos,ro:hasPart,_:ddspec) Type(_:ddspec,odm:DescriptiveDataSpecification) PropertyValue(_:ddspec,ro:hasPart,_:mappro) Type(_:mappro,opis:PropertyMappingSpecification) PropertyValue(_:ddspec,iao:isAbout,_:dt) Type(_:dt,odm:datatype) PropertyValue(_:mappro,iao:isAbout,_:prop) Type(_:prop,ssn:Property) PropertyValue(_:be,ssn:hasProperty,_:pro) Type(_:be,opis:behavioralEntity)
$\mathcal{VSO}$	output specification	PropertyValue(_:aos,ro:hasPart,_:odspec) Type(_:odspec,odm:OutputDataSpecification) PropertyValue(_:odspec,ro:hasPart,_:mapping) Type(_:be,mapping:semanticEntityMappingSpecification) PropertyValue(_:odspec,iao:isAbout,_:spdt) Type(_:spdt,opis:semanticPerceptionDatatype) PropertyValue(_:mapping,iao:isAbout, OR(_:be,_:sp,_:prop)) Type(_:so,opis:semanticProperty) PropertyValue(_:be,ssn:hasProperty,_:sp)
$\mathcal{VSE}$	operators	PropertyValue(id, ro:hasPart, _:ope) Type(_:ope,odm:operator) PropertyValue(_:ope,ro:roleOf,_:aim) Type(_:aim,odm:algorithmImplementation) PropertyValue(_:aim, obi:isConcretizationOf, _:algo) PropertyValue(_:ope, exact:hasInformation, _:pas)* Type(_:pas,odm:parameterSetting)* PropertyValue(_:aim, ro:hasQuality, _:par)* Type(_:par,odm:parameter)* PropertyValue(_:pas,iao:isQualitySpecificationOf,_:par)* PropertyValue(_:psp, iao:isAbout, _:par)* Type(_:psp,odm:parameterSpecification)*

Table 6.3 – Virtual sensor signature. Optional patterns are marked with *.

specification based on property mapping specifications, as defined in Table 6.2. Therefore, the semantic signature of PPVS is formally defined similarly to the virtual sensor signature without the $\mathcal{VSO}$ attribute. Lastly, Access Control Virtual Sensor (ACVS) are restricted to authorize personal information release. Thus, its signature can be formally defined similarly to virtual sensor signature without the $\mathcal{VSI}$ attribute, i. e. based solely on OPIS axioms (class, property object, and data property axioms).

CERTIFIED VIRTUAL SENSOR Virtual sensors can be considered trusted based on its verified certification. The *virtual sensor certification* acts as a certified stamp that assures the virtual sensor performs according to its semantic signature. The work involved to certify a virtual sensor can be based in several approaches, such as manual code verification, machine code verification, code identity verification, behavior heuristics, and so forth. In this manuscript, we refrain from defining these approaches, considering only their results. *Trust parties* should implement an independent certification process, providing two service interfaces: 1) virtual sensor certification request; and 2) virtual sensor certification verification.

This verification occurs during the virtual sensor installation, which queries the *trust party* about the certification verification using the provided public key for the virtual sensor. If valid, the system annotates persistently the virtual sensor using the opis:hasCertification annotation property in its semantic signature. For the matter of simplicity, we define an only-during-installation certification verification process. However, it is important to remark that more sophisticated mechanisms to update virtual sensor certification status or verify expired certifications can be implemented. The concept of certification is used in the PA-VSM to detect malicious inference intention in order to reinforce privacy.

6.2.3 Privacy Policy Condition

We propose to represent privacy policies based on ontology and perform decision evaluations based on results of SPARQL queries. In our approach, the PPC is based on three elements: information classification, privacy-preserving virtual sensor, and time interval. This condition is verified during the verification in the Privacy-aware Virtual Sensor Model (PA-VSM). We refer to the concerned individual who wants to enforce his/her privacy as *data provi-der*. PPC are individually defined by a *data provider*. A set of PPC forms a *privacy policy*.

The PPC is defined by extending the DUL concept of classification (*dul:classification*), which is related through the object property dul:isSettingFor to three classes: owl:Thing, dul:timeInterval, and dul:concept. Figure 6.8 depicts the semantic structure of PPC based on dul:classification structure. Since DUL does not define value

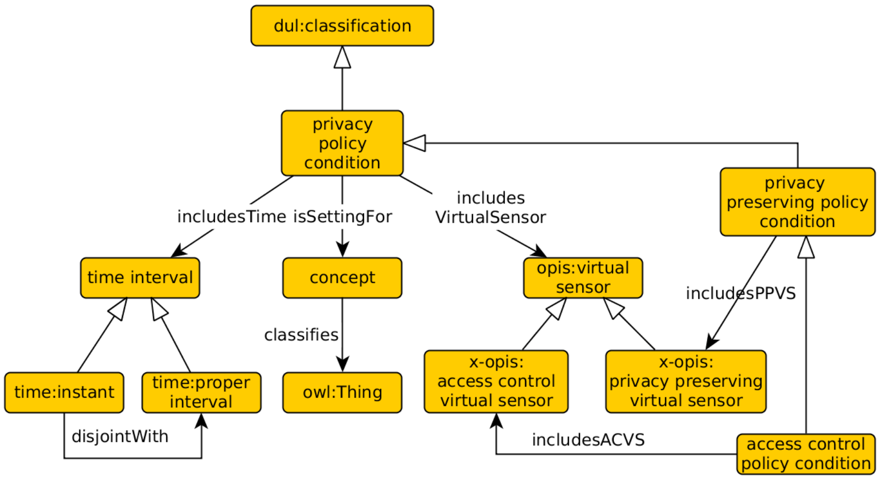

Figure 6.8 – Privacy policy condition structure

axiom	expression
includesInformationObject	$\sqsubseteq$ dul:isSettingFor$\sqsubseteq$dul:includesObject
includesVirtualSensor	$\sqsubseteq$ includesInformationObject
includesPrivacy-PreservingVirtualSensor	$\sqsubseteq$ includesVirtualSensor
includesAccessControl-VirtualSensor	$\sqsubseteq$ includesPrivacyPreservingVirtualSensor
privacyPolicyCondition	$\sqsubseteq$ dul:classification
	$\sqsubseteq \exists$ dul:includesTimes dul:timeInterval
	$\sqsubseteq \exists$ dul:isSettingFor (dul:concept $\sqcap \exists$ dul:classifies owl:Thing)
	$\sqsubseteq \exists$ includesVirtualSensor dul:concept
dul:timeInterval	$\equiv$ time:instant $\sqcap$ time:properInterval

Table 6.4 – Privacy policy condition definition

regions or unit of measurement, we import the OWLTime[2]. This allows defining privacy policy conditions based on instant (of any temporal kind, such as datetime), or interval (such as day, night, week). Each dul:classification is setting for a concept, and since we employ concepts to classify OPIS axioms, the PPC is linked to one or several OPIS axioms. Lastly, dul:classification is setting for (*dul:isSettingFor*) owl:Thing. In particular, we extend the object property dul:isSettingFor to specify the relation *includesVirtualSensor* between a PPC and a virtual sensor. The specialization of virtual sensor as defined in the previous Section is reflected in the structure of the PPC. Therefore, the Privacy-Preserving Policy Condition (PPPC) is a condition related only to PPVS and Access Control Policy Condition (ACPC) is related only to ACVS. Table 6.4 presents the formal definition of PPC classes and object properties.

2. `https://www.w3.org/TR/owl-time/` (accessed on 26/04/2017)

6.2.4 The Ontological Framework for Personal Information Classification

The concept described in this previous subsections constitute an ontological framework for personal information classification. That is possible because OPIS provides a foundation for classifying personal information, allowing extension of these entities to describe sub-domain and application dependent concepts. As explain previously, in our *privacy-by-policy* strategy, we provide a classification taxonomy based in the DOLCE-DnS UltraLite (DUL) that allows the *data provider* to classifies efficiently any kind of personal information defined in Personal Information Layer (PIL). Next, we extended the OPIS concept of virtual sensors to describe privacy-preserving technologies and access control mechanisms.

As PIL provides a structure for personal information classification, its classes are designed to be extended in sub-domain classes according to the application-dependent specificities. Similarly, virtual sensors are designed to be extended representing specific semantic perception processes, algorithm specifications, input specifications, output specifications, algorithm implementation, operators, and parameters. Despite the fact that Semantic Perception Layer (SPL) provides an initial set of algorithm specifications, each virtual sensor may have their own KDDM process and implementation that should be specified using SPL classes. Still, a common part of virtual sensor definition can emerge among installed virtual sensors, such as human activity perception objective, or algorithms for calculation of point of interest. PPVS and ACVS are specific cases of virtual sensors, as defined in Section 6.2.2. We define the set of extended PIL and PIL axioms as eXtended-OPIS (xOPIS).

Figure 6.9 depicts how these classes from OPIS are extended in xOPIS and how the PPC are related to the privacy-preserving virtual sensors, classification taxonomy, and consequently, to personal information. It is important to remark that ABox assertions, represented by dash-dotted boxes, play a role of instantiation of PPC classes. The *data provider* defines its *privacy policy* by instantiating individuals that concretely relates an instance of PPVS, depicted by the PPPC (concept_1, N, condition_1, M, ppvs_1), or an instance of ACVS, identified in the figure by the ACPC (concept_1, N, condition_1, P , acvs_1).

While the OPIS ontology and privacy policy ontology should be static, i. e. stay unchanged as a personal information classification and virtual sensor structure, the xOPIS ontology should be incremented each time a new virtual sensor signature is registered, defining its application specific OPIS axioms and virtual sensor definition. In addition, classification taxonomies can be registered along with new virtual sensors, or commonly shared among several virtual sensors.

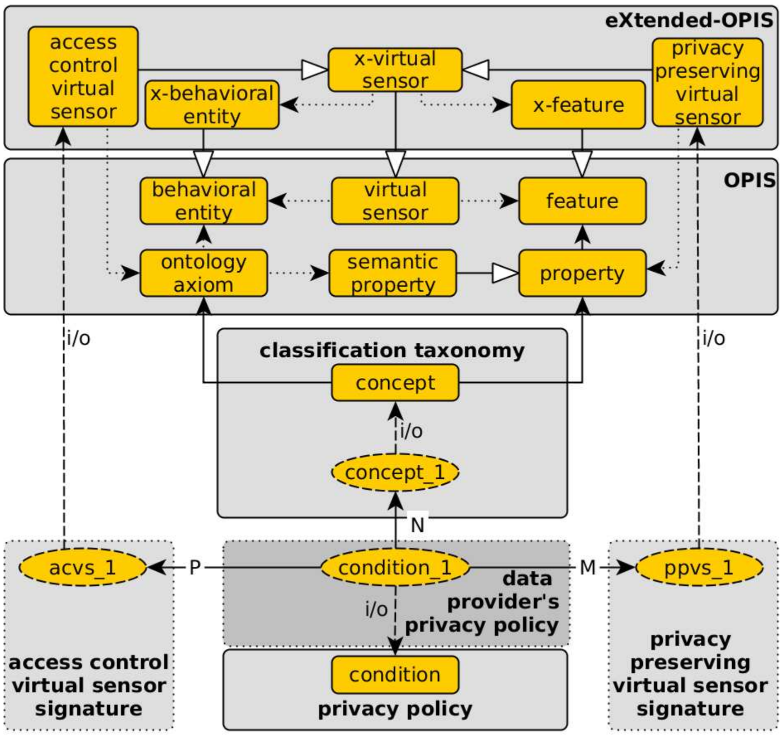

Figure 6.9 – Ontological framework for personal information classification. Dotted arrows represent property paths. Dash-dotted arrows labeled with i/o represent 'instance of' relations.

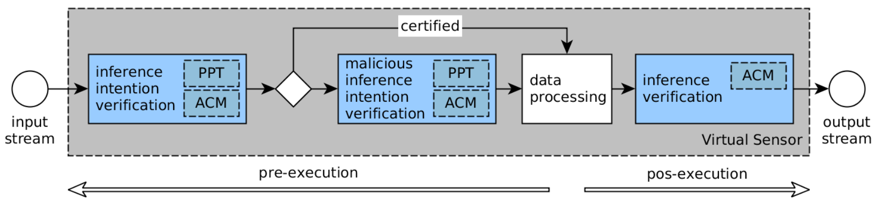

Figure 6.10 – Privacy-aware virtual sensor model

6.3 PRIVACY-AWARE VIRTUAL SENSOR MODEL

The Privacy-aware Virtual Sensor Model (PA-VSM) is a modular *privacy-by-design* model based on virtual sensors composed of a PEP of three verification steps which are performed before and after the KDDM process (data processing) step, as depicted in Figure 6.10. The pre-execution verifications aim to anticipate the KDDM inference intention based on virtual sensor signature in order to apply PPDMTs and ACMs over the input semantic data stream. Since our approach attempts to avoid privacy harm, i. e. an unintended inference of sensitive classified personal information, we provide an extra inference intention verification to detect malicious virtual sensor activities based on signature similarity. The post-execution verification aims to capture inferred classified information based on the KDDM output, making sure that only semantic perception is generated by virtual sensors.

6.3.1 Personal Semantic Data Stream

The concept of privacy is related to the data ownership. Therefore, the basic unit of collected data is tagged using a *data provider* identifier. The personal semantic data stream is, therefore, an observation value annotated with semantics that describes semantic observation type and *data provider* identification that allows reasoning about privacy policy conditions.

As OPIS is based on the Semantic Sensor Network Ontology (SSN-O), which provides a set of observation information, such as feature of interest, observed property, stimulus, quality of observation, sensing method, sampling time, and result time; we consider a subset of SSN-O annotations, defined as *semantic observation type*, that is used to characterize a semantic data stream. We define *semantic observation type* formally as follows:

Definition 11 (semantic observation type) *Let $\mathcal{X}$ be an ontology for personal information and virtual sensors, such as OPIS. A semantic observation type is a tuple **sot** = <foi, prop, obs, dtt> iff:*
— $\llbracket$ *Type(foi, ssn:featureOfInterest)* $\rrbracket_{\mathcal{X}}$ *is satisfied;*
— $\llbracket$ *Type(prop, ssn:property)* $\rrbracket_{\mathcal{X}}$ *is satisfied;*
— $\llbracket$ *Type(obs, dul:region)* $\rrbracket_{\mathcal{X}}$ *is satisfied;*
— $\llbracket$ *Type(dtt, odm:datatype)* $\rrbracket_{\mathcal{X}}$ *is satisfied;*

SSN-O does not define regions and unit of measurement, suggesting that that semantics should be defined or imported from another ontology in order to provide format and value space for *sensor output* (ssn:sensorOutput). It defines *observation value* ($\sqsubseteq$ dul:region $\sqsubseteq$ ssn:observationValue) as regions for sensor outputs. Thus, we complement this semantics by employing the OntoDM class of *datatype* (odm:datatype) to define the format of ssn:observationValue, as defined in 5.8.3. This allows compatibility to the format dataset are defined to algorithms in OntoDM and, consequently, virtual sensors. The odm:datatype is related to ssn:observationValue through the IAO object property of *is about* (iao:isAbout) to ssn:ObservationValue.

These values are aggregated as time-series observations of a specific *semantic observation type*. Virtual sensors are notified about data streams of a specific semantic observation type, which are consistent RDF graphs that represent a set of observation annotated. We formally defined the *semantic sensor observation* as follows:

Definition 12 (semantic sensor observation) *Let $\mathcal{X}$ be an ontology for personal information and virtual sensors, such as OPIS. A semantic sensor observation is a consistent RDF graph $\mathcal{RSS}$ = <$\mathcal{OBSV}$:[sot][t]> iff: $\mathcal{OBSV}$ is an RDF graph where $\exists$ (v, rdf:type, obs) $\in \mathcal{OBSV}$ iff $\mathcal{OBSV}$ is consistent over $\mathcal{X}$, and indexed by a semantic observation type sot and a timestamp t.*

The unit of observation is, therefore, an RDF graph containing RDF triples with assertions about the set of *semantic observation type* and *observation value*. The concept of consistency of RDF graph over a specific ontology is important in this context because it is not possible to reason over inconsistent RDF graphs. In addition, this guarantees that the format and semantics defined in the ontology for personal information and virtual sensors $\mathcal{X}$ are verified and valid.

The IoT sensing is a continuous stream of data that notifies virtual sensors when input stream conditions are reached, such as window type, window size or interval, and *semantic observation type*. This set of *semantic sensor observation* is called Personal RDF Stream Sample (PRSS) and it is annotated, among the up-mentioned information, with the *data provider*'s identifier (**pid**). For the matter of simplicity, we won't incorporate technical details about the input stream conditions, refraining ourselves in this chapter to the concept of a finite set of *semantic sensor observation*. The technical details about this are explained in the next chapter. We formally define PRSS as follows:

Definition 13 (personal RDF stream sample) *A personal data stream sample is a set of semantic sensor observation* $\mathcal{OBSV}_{\mathtt{pid}} = \{<\mathcal{OBSV}>\}$, *where* ***pid*** *denotes an individual's semantic sensor observation stream.*

This PRSS is the data stream that will be verified and processed by virtual sensors. PPDMTs transform PRSSs in order to minimize data utility, such as through *l*-diversity or other obfuscation techniques. It should be noticed that OPIS specifies virtual sensor inputs considering both physical sensor observations and virtual sensor observations. In the former, *semantic observation types* refer to real World entities, which can be observed using physical sensors, observation value regions, and sensor output datatypes. In the latter, *semantic observation types* correspond to semantic perception of class and object property axioms, object property assertions, and data property assertions. Our approach allows considering semantic perceptions as observations. As a consequence, PPDMTs can be applied over semantic perceptions. Therefore, higher-level privacy preservation can be achieved, such as *l*-diversity applied to types of point of interest (instead of raw geographic location points).

6.3.2 Privacy Enforcement Process

The privacy enforcement process of PA-VSM is presented in Figure 6.11 using BPMN 2.0 notation. The pre-execution verification has three main objectives: i) prevent virtual sensors to execute data processing which results are not authorized according to the ACMs defined in individual's PPC; ii) decrease data utility selectively by applying PPDMTs that minimize the chances of extracting results that are not authorized according to individual's PPC; iii) identify and prevent malicious non-certified virtual sensors to perform data processing in PRSS. The post-execution verification has two main objectives: i) verify the semantic observation value consistency based on the semantic perception datatype; and ii) prevent unauthorized output release according to the ACMs defined in individual's PPC.

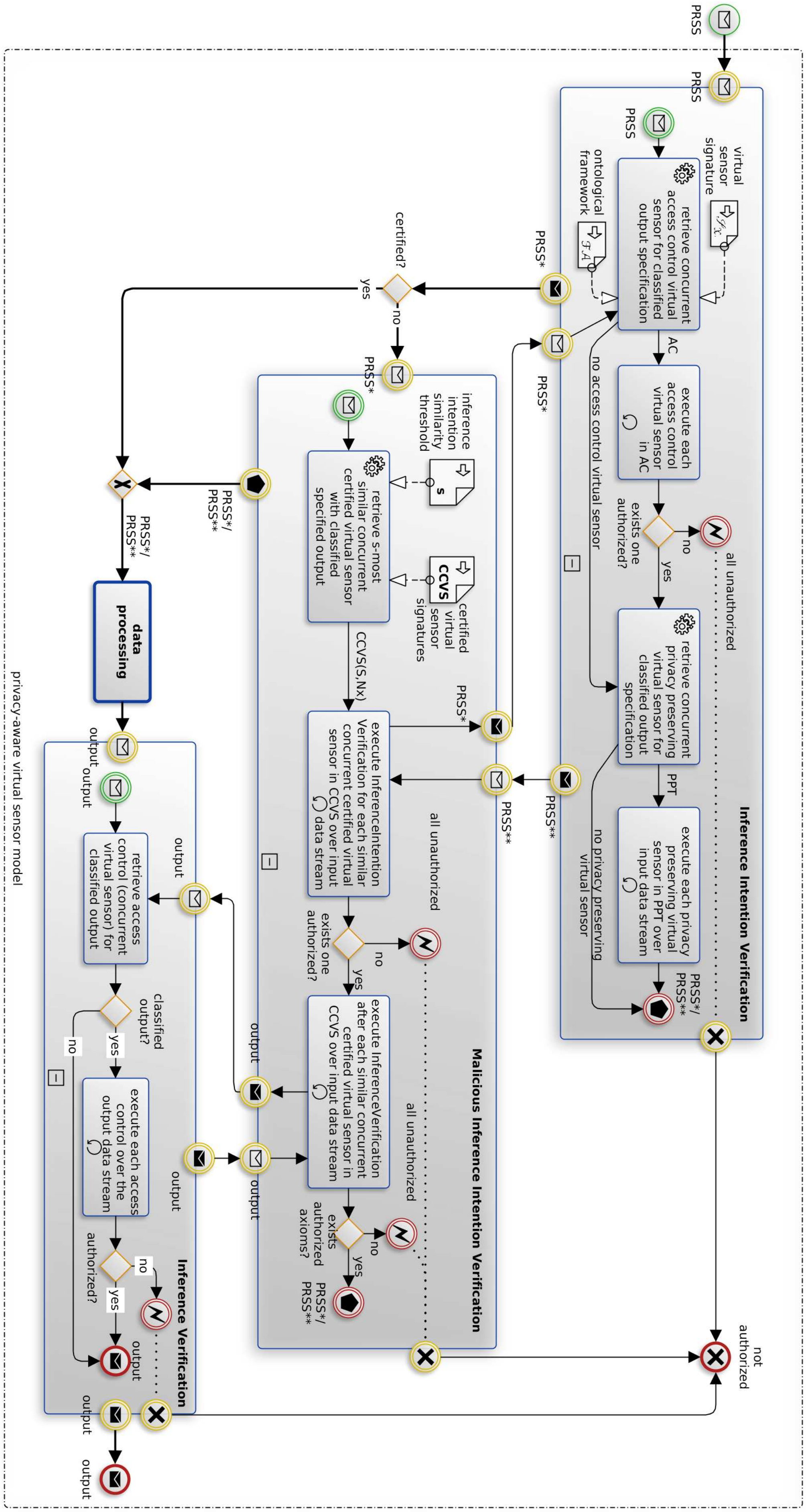

Figure 6.11 – Privacy enforcement process

6.3.2.1 *Inference Intention Verification*

The advantage of having a semantic signature for virtual sensor, besides the KDDM provenance, is the possibility to perform SPARQL queries over a set of signatures that share the same ontological framework and, consequently, compared each other. The Personal Information Layer (PIL) provides the cognitive base for individuals to understand and classify their own data and information, defining PPC that selectively apply Privacy-Preserving Data Mining Techniques (PPDMTs) and Access Control Models (ACMs). On the other hand, the Semantic Perception Layer (SPL) offers the underlying structure to classify and compare Knowledge Discovery and Data Mining (KDDM) that are implemented using virtual sensors.

In our approach, Privacy Policy Conditions (PPCs) are represented by assertions that relate personal information (OPIS class or object property axioms); time interval (or instant); and Privacy-Preserving Virtual Sensor (PPVS) or Access Control Virtual Sensor (ACVS). This structure relates the behavioral context where the personal information is used or controlled, and the informational context on which this information is inferred. The temporal unit can be adapted to express starting instant, data interval, or period of PPC validity.

In Access Control Policy Condition (ACPC), ACVS may prevent PRSS to be released, while in Privacy-Preserving Policy Condition (PPPC), PPVS decreases data quality aiming to minimize the chances or accuracy of sensitive information to be inferred from PRSS. Therefore, virtual sensor KDDM step should only be executed if there is any chance of authorized output. Several ACMs are proposed and can be implemented as ACVS, such as Role-Based Access Control [160] and Attribute-Based Access Control, which has been broadly adopted along with the XACML [212].

In Privacy-Preserving Policy Condition (PPPC), PPVS should be executed before the virtual sensor KDDM step in order to transform the PRSS and minimize the chances or the accuracy of sensitive information, for example, ϵ-differential privacy or k-anonymization.

Algorithm 2 presents the implementation of the *inference intention verification* step. The input is formed by the ontology for personal information $\mathcal{X}$, semantic signature for virtual sensor in execution $\mathscr{S}_{\mathcal{X}}$, classification taxonomy $\mathcal{C}$, direct classification $\mathcal{D}$, privacy policy conditions $\mathcal{P}$, personal identifier **pid**, and PRSS $\mathcal{OBSV}_{\mathtt{pid}}$.

The function **ExecuteVS** has as input the virtual sensor id, the PRSS, and an optional signature referring the virtual sensor from which the access control was called. In line 7, if the **ExecuteVS** function returns something, there is at least one possibility that the virtual sensor will result in authorized content, and, therefore, should proceed with the virtual sensor execution. While in line 19, it returns the result of PPDMTs over PRSS for each *semantic observation type* (sot).

Additionally, the functions **RetrieveACVSForVS** (line 3) and **RetrievePPVSForVS** (line 16) returns SPARQL-DL queries. As upmentioned, our approach relies on the SPARQL queries to evaluate PPCs and retrieve PPDMTs. The function **Sparql($\mathcal{O}$,q)** returns a set of

Input : $\mathcal{X}$, $\mathscr{S}_{\mathcal{X}}$, $\mathcal{C}$, $\mathcal{D}$, $\mathcal{P}$, IDS
$\mathcal{X}$: ontology for representation of personal information and virtual sensors
$\mathscr{S}_{\mathcal{X}}$: virtual sensor signature
$\mathcal{C}$: classification taxonomy
$\mathcal{D}$: direct classification
$\mathcal{P}$: privacy policy conditions
pid: personal identifier
$\mathcal{OBSV}_{\mathtt{pid}}$: personal RDF stream sample
Output : personal RDF stream sample

begin
  $\mathcal{FA} = \mathcal{X} \cup \mathcal{C} \cup \mathcal{D} \cup \mathcal{P}$
  `// Output access control based on classified output specification`
  AC ← **Sparql**($\mathcal{FA}$,**RetrieveACVSForVS**($\mathscr{S}_{\mathcal{X}}.\mathcal{CID}$.id, pid))
  all_unauthorized← true
  foreach *<ac,axiom> in AC* **do**
    sot ← <opis:behavioralEntity, opis:semanticProperty, opis:ontologyAxiom, opis:semanticPerceptionDatatype>
    obs[sot,timestamp] ← **axiom**
    if ***ExecuteVS****(ac,obs,$\mathscr{S}_{\mathcal{X}}$)!=nil* **then**
      all_unauthorized← false
      break
    end
  end
  if *all_unauthorized* **then**
    return nil `// interrupt data stream if all output is unauthorized`
  end
  `// Privacy-preserving for input based on output specification and direct classification`
  PPT ← **Sparql**($\mathcal{FA}$,**RetrievePPVSForVS**($\mathscr{S}_{\mathcal{X}}.\mathcal{CID}$.id, pid))
  `// PPT={<foi,prop,obs,dtt,ppt>}`
  foreach *<foi,prop,obs,dtt,$\mathscr{P}_{\mathcal{X}}$> in PPT* **do**
    sot ← <foi,prop,obs,dtt>
    $\mathcal{OBSV}_{\mathtt{pid}}$[sot] ← **ExecuteVS**($\mathscr{P}_{\mathcal{X}}.\mathcal{CID}$.id,$\mathcal{OBSV}_{\mathtt{pid}}$[sot],$\mathscr{S}_{\mathcal{X}}$)
  end
  return $\mathcal{OBSV}_{\mathtt{pid}}$
end

Algorithmus 2 : Inference Intention Verification variable binding for the evaluation mappings of query **q** over the ontology $\mathcal{O}$, which is traditionally implemented in OWL reasoners, such as in the Apache Jena framework [3].

We present SPARQL-DL queries using tables describing clauses types (select, where, sub-query), ontological abstraction levels (Tbox, RBox, ABox), SPARQL-DL query atoms and ids. More complex queries are based on subqueries that extract information about virtual sensors from their signatures, such as output specification and input specification. In order to simplify and reuse SPARQL-DL queries, we define those queries as functions which variables can be replaced and used as input and output parameters.

clause	type	#id	query
where	TBox/ABox	TA1	Type(**?IVirtualSensor**,opis:virtualSensor)
		TA2	DirectType(_:spp,opis:SemanticPerceptionProcess)
		TA3	Type(_:alg,obi:algorithm)
		TA4	Type(_:ods,odm:outputDataSpecification)
		TA5	Type(?**IDatatype**,opis:semanticPerceptionDatatype)
		TA6	Type(_:semap,opis:semanticEntityMappingSpecification)
		TA7	Type(**?IAxiom**,opis:ontologyAxiom)
		TA8	Type(_:FoI,ssn:FeatureOfInterest)
		TA9	Type(_:Property,ssn:Property)
	ABox	A2	PropertyValue(**?IVirtualSensor**,obi:isConcretizationOf,_:spp)
		A3	PropertyValue(_:spp,ro:hasPart,_:alg)
		A4	PropertyValue(_:alg,ro:hasPart,_:ods)
		A5	PropertyValue(_:ods,ro:hasPart,_:semap)
		A6	PropertyValue(_:semap,iao:isAbout,**?IAxiom**)
		A7	PropertyValue(**?IAxiom**,opis:hasIRI,OR(_:IFoI,_:IProperty))

Table 6.5 – **OutputSpecVS**: Query definition to retrieve virtual sensor output specification

For example, the sub-query **OutputSpecVS** presented in Table 6.5 retrieves the virtual sensor output specification based on the virtual sensor id (?IVirtualSensor), personal information (?IAxiom) and datatype (?IDatatype). As any SPARQL query, these variables can have fixed values or *evaluation mappings*. If a value is defined for ?IVirtualSensor, for instance, only *evaluation mappings* related to this virtual sensor is retrieved, if they exist in the RDF graph. Ontological abstraction levels are grouped to facilitate the visualization of class assertions (TBox/ABox) and object property assertions (ABox) in *where* clause. Conversely, the sub-query **InputSpecVS** presented in Table 6.6 retrieves the virtual sensor input specification based on the virtual sensor id(?IVirtualSensor), feature of interest (?IFoI), property (?IProp), and datatype (?IDatatype). These queries are designed based on the virtual sensor signature, which is defined in 6.3.

Based on **OutputSpecVS**, the SPARQL-DL query to retrieve ACVSs for classified personal information **RetrieveACForVS** is defined in Table 6.7. Given a virtual sensor (?IVirtualSensor), the sub-query SQ1 returns output specification (?IAxiomOut, _:IDatatypeOut), linking the semantic perception (?IAxiomOut) to classification concept (_:ICon-

3. https://jena.apache.org/ (accessed on 26/04/2017)

clause	type	#id	query
where	TBox/ ABox	TA1	Type(**?IvirtualSensor**,opis:virtualSensor)
		TA2	DirectType(_:spp,opis:SemanticPerceptionProcess)
		TA3	Type(_:algo,obi:algorithm)
		TA4	Type(_:dds,odm:descriptiveDataSpecification)
		TA5	Type(?**IDatatype**,odm:datatype)
		TA6	Type(_:map,odm:mappingSpecification)
		TA7	Type(?**IProperty**,ssn:Property)
		TA8	Type(?**IFoI**,ssn:FeatureOfInterest)
	ABox	A1	PropertyValue(_:vs,obi:isConcretizationOf,_:spp)
		A2	PropertyValue(_:spp,ro:hasPart,_:algo)
		A3	PropertyValue(_:alg,ro:hasPart,_:dds)
		A4	PropertyValue(_:dds,iao:isAbout,?**IDatatype**)
		A5	PropertyValue(_:dds,ro:hasPart,_:map)
		A6	PropertyValue(_:map,iao:isAbout,?**IProperty**)
		A7	PropertyValue(?prop,ssn:isPropertyOf,?**IFoI**)

Table 6.6 – **InputSpecVS**: Query definition to retrieve virtual sensor input specification

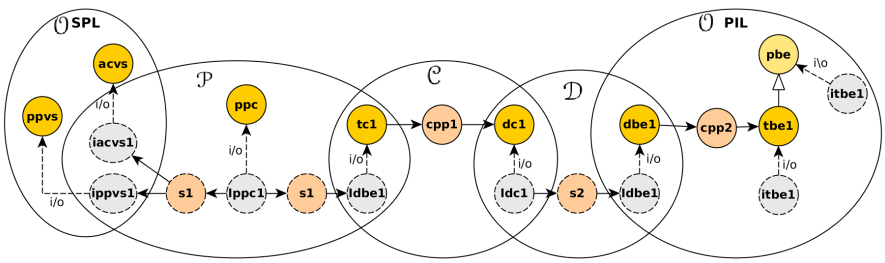

Figure 6.12 – Privacy policy conditions

cept) through a transversal classification (sub-query SQ2). Thus, the ontological framework described in Section 6.2.4 can be taken into account to retrieve ACVS. The ACPC is expressed relating the classification concept (_:IConcept) to the classification condition (_:ICondition), and, consequently, to a ACVS (?IACVS). Figure 6.12 depicts this property path between the semantic perception (*itbe1*) and the ACVS (*iacvs1*).

The semantic perception (*itbe1*) represents the personal information, which can be an axiom assertion (class, object property, or data property assertion). In the case depicted in figure 6.12, the ACVS assertion *Iacvs1* encodes the ACVS id.

The sub-query SQ1 retrieves all output specification (?IAxiomOut, _:IDatatypeOut) from a given virtual sensor (?IVirtualSensor), binding these output axioms (?IAxiomOut) to some classification concept through the **TransversalClassification** sub-query (SQ2), as defined in Table 6.1. The classification concept is, then, related to conditions and, consequently, to the ACVS (A2,A3), according to the PPC definition (Table 6.4).

Similarly, the SPARQL-DL query to retrieve PPVS **RetrievePPVSForVS** is also based on property path that relates a personal information and a PPVS, as defined in Table 6.8. PPVSs transform the input (PRSS) aiming to prevent the inference of a classified personal

clause	type	#id	query
sub-query	-	SQ1	**OutputSpecVS**(**?IVirtualSensor**,**?IAxiomOut**,_:IDatatypeOut)
		SQ2	**TransversalClassification**(_:IConcept,**?IAxiomOut**)
where	TBox/ ABox	TA1	Type(**?IACVS**, x-opis:AccessControlVirtualSensor)
		TA2	Type(_:IDatatypeOut, opis:SemanticPerceptionDatatype)
	ABox	A2	PropertyValue(_:IConcept,dul:hasSetting,_:ICondition)
		A3	PropertyValue(_:ICondition,opis:includesACVS,**?IACVS**)

Table 6.7 – RetrieveACVSForVS: Query definition to retrieve access control virtual sensor

information by a virtual sensor (?IVirtualSensor). Therefore, in order to retrieve those PPVSs (?IPPVS) that should be executed, an extra property path pattern to retrieve all PPVSs related to the virtual sensor (?IVirtualSensor) is realized by associating their inputs (SQ2, SQ3). In Figure 6.12, a classified personal information (*itbe1*) is related through a PPC to a PPVS (*ippvs1*).

clause	type	#id	query
select	ABox	A1	**?IFoIIn**, **?IPropertyIn**, **?IDatatypeIn**, **?IPPVS**
sub-query	-	SQ1	**OutputSpecVS**(**?IVirtualSensor**):_:IAxiomOut,_:IDatatypeOut
		SQ2	**InputSpecVS**(**?IVirtualSensor**):**?IFoIIn**,**?IPropertyIn**,**?IDatatypeIn**
		SQ3	**InputSpecVS**(**?IPPVS**):**?IFoIIn**,**?IPropertyIn**,**?IDatatypeIn**
		SQ4	**TransversalClassification**(_:IConcept):_:IAxiomOut
where	TBox/ ABox	TA1	Type(**?IPPVS**, x-opis:PrivacyPreservingVirtualSensor)
	ABox	A2	PropertyValue(_:IConcept,dul:hasSetting,_:ICondition)
		A3	PropertyValue(_:ICondition,opis:includesPPVS,**?IPPVS**)

Table 6.8 – RetrievePPVSForVS: Query definition to retrieve privacy-preserving virtual sensor

6.3.3 Malicious Inference Intention Verification

In the case of the non-certified virtual sensor, its semantic signature is not verified and, therefore, may not represent its real inference intention. The possibility of a malicious activity, when inference specification is intentionally different from the data processing implementation, is a weak chain in our privacy verification. Some alternatives to address this issue are possible. The simplest solution corresponds to prevent completely the execution of non-certified virtual sensors. This extreme solution does not represent the reality in deployed systems where virtual sensors are deployed in the Cloud in a time-to-market fashion. Thus, an automatic verification to minimize the chances of malicious inference intention is needed. Techniques to identify execution traces [213], which consists in analyze transactional data from algorithm executions to detect and identify application implementation; or techniques to detect malicious activities based on crowdsourcing reputation systems [214].

In our approach, our semantic signature for virtual sensor model provides a formalization how to represent the result of these tech-

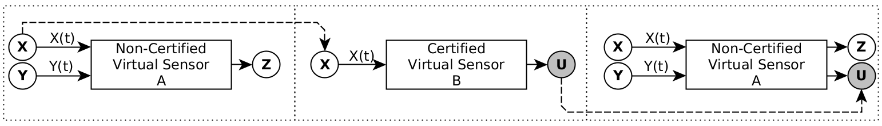

Figure 6.13 – Malicious Inference Intention Logic

niques in order to compare algorithms using the same semantics. We propose to compare the similarity between pairs of *concurrent virtual sensors* which share a classified output subset. The logic behind the malicious inference intention step relies on the similarity to certified virtual sensors that may use the same input (or a subset of it) and infer a classified information. Figure 6.13 depicts this logic, representing how a non-certified virtual sensor **A** can be related to a classified information **U** when compared to a similar concurrent virtual sensor **B**. However, this verification goes beyond the previous static inference intention verification. In this case, given a PRSS, the similar concurrent virtual sensor is executed in order to verify if the input **X(t)** indeed results in classified information **U**.

Concurrent virtual sensors share an input specification subset and it is defined formally as follows:

Definition 14 (Concurrent virtual sensor) *Be $\mathcal{X}$ an ontology for representation of personal information and virtual sensor, $\mathcal{OBSV}^{\mathcal{S}_\mathcal{X}}_{\mathtt{pid}}$ a personal RDF stream sample input of virtual sensor $\mathcal{S}_\mathcal{X}$. A **concurrent virtual sensor** $\mathcal{N}_\mathcal{X}$ is a virtual sensor which input $\mathcal{OBSV}^{\mathcal{N}_\mathcal{X}}_{\mathtt{pid}} \subseteq \mathcal{OBSV}^{\mathcal{S}_\mathcal{X}}_{\mathtt{pid}}$*

The SPARQL-DL query to retrieve *concurrent virtual sensors* is defined in Table 6.9. In this query, two graphs G1 and G2 are constructed based on the input specification of the virtual sensor in execution (?IVirtualSensor) and of *concurrent virtual sensors* (?IConVS), respectively. The graph construction defines 4 RDF triples (A1-A4, A5-A9) to be compared. The input specification is retrieved using the **InputSpecVS** sub-query (SQ1, SQ2). Finally, G1 are compared to G2, such that G1 minus G2 results in nothing (A8).

graph	clause	type	#id	query
G1	construct	ABox	A1	**?IConVS** a opis:virtualSensor
			A2	_:ICFoI a ssn:FeatureOfInterest
			A3	_:ICProp a ssn:Property
			A4	_:ICDttype a odm:Datatype
	sub-query	-	SQ1	**InputSpecVS(?IConVS**,_:ICFoI,_:ICProp,_:ICDttype)
G2	construct	ABox	A5	**?IVirtualSensor** a opis:virtualSensor
			A6	_:IFoI a ssn:FeatureOfInterest
			A7	_:IProp a ssn:Property
			A9	_:IDttype a odm:Datatype
	sub-query	-	SQ2	**InputSpecVS(?IVirtualSensor)**:?IFoI,?IProp,?IDttype
-	filter	ABox	A8	G1 not exists G2

Table 6.9 – **ConcurrentVS**: Query definition to concurrent virtual sensors

If we consider a large semantic signature knowledge base, the comparison algorithm complexity can be unfeasible for data stream processing. Therefore, we restrict the search space for only *concurrent virtual sensors* that are certified and which output specification contains classified personal information. In addition, aiming to limit the number of possible virtual sensors to compare, we employ the concept of similarity between virtual sensor signatures. An extensive sort of semantic measures is proposed [215].

Traditionally, this is an NP-complete problem which demands a considerable processing power. In the next chapter, we describe the architectural design to address this issue. For now, we simply use the similarity result as a constraint for the SPARQL-DL query **SimilarConcurrentCertifiedVS** defined in Table 6.10 which retrieves all *concurrent virtual sensors* (?IConVS) with a minimum similarity to the virtual sensor in execution (?IVirtualSensor).

clause	type	#id	query
sub-query	-	SQ1	**ConcurrentVS(?IVirtualSensor,?IConVS)**
where	TBox/ ABox	TA1	Type(_: VSSim, iao:dataItem)
		TA2	Type(_:simCalc, obi:similarityCalculation)
		TA3	Type(_:realDataType, odm:realDatype)
	ABox	A1	PropertyValue(**?IConVS**, x-opis:hasCertification, _:Cert)
		A2	PropertyValue(**?IConVirtualSensor**, ro:hasPart, _:VSSim)
		A3	PropertyValue(**?IVirtualSensor**, ro:hasPart, _:VSSim)
		A4	PropertyValue(_:VSSim, obi:isSpecifiedOutputOf, _:simCalc)
		A5	PropertyValue(_:realDataType, iao:isAbout, _: VSSim)
		A6	PropertyValue(_:VSSim, _:dataProperty, **?similarity**)
		A7	Filter(**?similarity** >= **?similarityThreshold**)
	RBox	R1	DataProperty(_:dataProperty)

Table 6.10 – SimilarConcurrentCertifiedVS: Query definition to retrieve similar concurrent virtual sensors

Concurrent virtual sensors are retrieved using the **ConcurrentVS** sub-query (SQ1) and need to have a certification (A1). Since several semantic similarity and relatedness measures are possible, the similarity measure is a data item (TA1) defined using OBI class of similarity calculation (TA2,A3) which result is defined using OntoDM class of real datatype (TA3,A4). The virtual sensor in execution and the concurrent virtual sensor have commonly the similarity measure (A1, A2). A data property related to the semantic measure is then limited to the similarity threshold (A5,R1,A6).

Therefore, based on these definitions, the detection of malicious inference intention is defined using the **SimilarConcurrentCertifiedVS** and the algorithms **InferenceIntentionVerification** and **InferenceVerification**, as presented in Algorithm 3.

A set of similar concurrent certified virtual sensors is retrieved using the **SimilarConcurrentCertifiedVS** and a similarity threshold **s** (line 2). For each similar concurrent certified virtual sensor, its inference intention is verified using the first step function **InferenceIntentionVerification** (line 5). The input PRSS is replicated (line 3),

Input : $\mathcal{X}$, $\mathcal{N}_{\mathcal{X}}$,s, $\mathcal{C}$, $\mathcal{D}$, $\mathcal{P}$, IDS
$\mathcal{X}$: ontology for representation of personal information and virtual sensors
$\mathcal{N}_{\mathcal{X}}$: virtual sensor signature
$\mathcal{C}$: classification taxonomy
$\mathcal{D}$: direct classification
$\mathcal{P}$: privacy policy conditions
IDS: input semantic data stream sample $\mathrm{IDS}=\{<\mathrm{IDS}>_{T_1},\ldots,<\mathrm{IDS}>_{T_n}\}$,
where $<\mathrm{IDS}>_{T_i}=\{(\mathrm{obsValue},\mathrm{timeSampling})\}$ and $T_i=<\mathrm{FoI},\mathrm{Prop},\mathrm{Datatype}>$
Output : output semantic data stream sample

1 **begin**
2 CCVS$^s_{\mathcal{N}_{\mathcal{X}}}$ = **Sparql**($\mathcal{X}$, **SimilarConcurrentCertifiedVS**($\mathcal{N}_{\mathcal{X}}$,s)) // CCVS$^s_{\mathcal{N}_{\mathcal{X}}}$={<$\mathcal{S}_{\mathcal{X}}$>}
3 $\mathrm{IDS}^* = \mathrm{IDS}$
4 **foreach** $\mathcal{S}_{\mathcal{X}}$ *in CCVS*$^s_{\mathcal{N}_{\mathcal{X}}}$ **do**
5 IDS* = **InferenceIntentionVerification**($\mathcal{X}$,$\mathcal{S}_{\mathcal{X}}$.$\mathcal{CID}$.id,$\mathcal{C}$,$\mathcal{D}$,$\mathcal{P}$,IDS*)
6 **if** *IDS*=nil* **then**
// if stream is not authorized, return nil
7 return nil
8 **end**
9 ODS←**ExecuteVS**($\mathcal{S}_{\mathcal{X}}$.$\mathcal{CID}$.id,IDS*,$\mathcal{N}_{\mathcal{X}}$)
10 ODS←**InferenceVerification**($\mathcal{X}$,$\mathcal{S}_{\mathcal{X}}$.$\mathcal{CID}$.id,$\mathcal{C}$,$\mathcal{D}$,$\mathcal{P}$,ODS)
11 **if** *ODS<Axioms>=nil* **then**
// if output is not authorized, return nil
12 return nil
13 **end**
14 **end**
15 **return** IDS*
16 **end**

Algorithmus 3 : Malicious Inference Intention Verification

and transformed through this verification, which may apply ACMs or PPDMTs according to the retrieved concurrent virtual sensors. if some ACPC in **InferenceIntentionVerification** do not authorize the PRSS, the malicious inference intention verification returns nothing (line 7). If IDS is still authorized, the similar concurrent certified virtual sensor is executed (line 9) and its output is verified using the **InferenceVerification** to guarantee that no classified information can be extracted from the input PRSS.

6.3.4 Inference Verification

Ultimately, the pre-execution steps aim to prevent a malicious algorithm to extract unintended personal information, trying to anticipate its KDDM process. We propose an approach preventive to detect malicious inference intention. Therefore, after the KDDM execution, there is still the need to verify its output because of the possibility on which a virtual sensor can proceed its KDDM step if it exists authorized output – there is no ACPC related to it – or private preserved output – there is one or more PPPCs related to it.

The result of a virtual sensor must be a **semantic perception** as defined in our approach in chapter 5. As previously mentioned, this mechanism assures the output data type and semantics, which allows implementing a verification based on PPC. The *semantic perception datatype* is defined in OPIS and has three attributes: *axioms*, *object property assertions*, and *data property assertions*. The *axioms* attribute refers to a set of IRI, while the *object property assertions* attribute and *data property assertions* attribute to refer to set of RDF triples.

```
Input   : 𝒳, 𝒮_𝒳, s, 𝒞, 𝒟, 𝒫, IDS
          𝒳: ontology for representation of personal information and virtual sensors
          𝒮_𝒳: virtual sensor signature
          𝒞: classification taxonomy
          𝒟: direct classification
          𝒫: privacy policy conditions
          ODS: output semantic data stream sample ODS = {{<axiom>}_Axioms,
          {<classAxiom,objectPropertyAxiom,classAxiom>}_ObjectPropertyAssertions,
          {<classAxiom,dataPropertyAxiom,obsValue,Datatype>}_DataPropertyAssertions,
          timeSampling}
Output  : output semantic data stream sample
begin
    𝓕𝓐 = 𝒳 ∪ 𝒞 ∪ 𝒟 ∪ 𝒫
    A ← ODS<Axioms> // Verify axiom declarations
    A* ← nil
    foreach axiom in A do
        ac = Sparql(𝒳, RetrieveACVSForAxiom(𝓕𝓐,axiom))
        if (ac and AccessControl(ac,axiom,𝒮_𝒳)) then
            A* ←<axiom>
        end
    end
    OP ← ODS<ObjectPropertyAssertions> // Verify object property assertions
    OP* ← nil
    foreach <subject,predicate,object> in OP do
        if {<subject>,<predicate>,<object>} in A* then
            OP* ←<subject,predicate,object>
        end
    end
    DP ← ODS<DataPropertyAssertions> // Verify data property assertions
    DP* ← nil
    foreach <subject,predicate,data,datatype> in DP do
        if {<subject>,<predicate>} in A* then
            if ConsistentDatatype(data,datatype) then
                DP* ←<subject,predicate,data>
            end
        end
    end
    return {A*_Axioms,OP*_ObjectPropertyAssertions,DP*_DataPropertyAssertions}
end
```

Algorithmus 4 : Inference verification

It is important to remark that the OPIS class of *semantic perception datatype* inherents from OntoDT a flexibility to restrict and redefine aggregate datatypes. Thus, it is possible to restrict the IRI value space to allow only a subset of IRI. In this manuscript, we do not restrict the IRI value space in the ontology. We do it by defining *semantic perception observation type*:

Definition 15 (semantic perception observation type) *Let $\mathcal{X}$ be an ontology for personal information and virtual sensors, such as OPIS. A semantic perception observation type is a tuple* ***spot*** = <$\mathcal{BE}$, $\mathcal{OBA}$, $\mathcal{DPA}$> *iff:*

— $[\![$ *Type(be, opis:behavioralEntity)* $]\!]_{\mathcal{BE}}$ *is satisfied;*

— $[\![$ *PropertyValue(be1, prop, be2)* $]\!]_{\mathcal{OBA}}$ *is satisfied, where be1, prop, be2* $\in \mathcal{BE}$*;*

— $[\![$ *PropertyValue(be, aprop, l)* $]\!]_{\mathcal{DPA}}$ *is satisfied, where be, aprop* $\in \mathcal{BE}$*, and* $l \in \mathbf{L}$*;*

In other words, the inference output of virtual sensors is a list of IRI that are used in the semantic assertions (*object property assertions* and *data property assertions*). Based on this definition, the Algorithm 4 is defined to verify these three output attributes. The set of axiom declarations **A** (line 3) is verified against the ACPCs, adding authorized axioms to a new set **A***. Next, the set of RDF triple of object property assertions (line 11) is verified to guarantee that only declared axioms are asserted. At last, the set of RDF triple of data property assertions (line 18) is verified to guarantee that only declared axioms are asserted and that the datatype is consistent with the declared output datatype.

In the end, the authorized inference output is returned, and, consequently released to access from other virtual sensors or upper-level layers, such as service layer or application layer.

6.4 CONCLUSION

We presented in this chapter a privacy mechanism for the IoT sensing based on a *privacy-by-policy* and *privacy-by-design* strategy.

The *privacy-by-policy* part of the strategy is achieved through an ontological framework that allows specifying personal information, classification taxonomies, and privacy policy conditions from a cognitive perspective, using the DOLCE-DnS UltraLite (DUL) conceptual framework. This aspect of our approach addresses the understanding obstacle that IoT sensing poses for non-technical *data providers*, supporting personal information classification and privacy policy definition. Part of this issue is caused by the lack of common semantics to compare personal information, which is supplied by OPIS. Additionally, the capacity to extend OPIS provides a flexible semantics for virtual sensor developers to represent specific personal information used or produced by their virtual sensors. Due to OPIS structure, this extension provides means to reason about classification conditions and to apply the concept of *transversal classification* to reach concepts that are not directly classified but are somehow related to a classified information.

On another hand, our *privacy-by-design* is based on the capacity to encapsulate KDDM processes into virtual sensors and represent their inputs, outputs, objectives, specifications, algorithms, implementations, and parameters using semantic concepts which allow comparing virtual sensors and creating mechanisms to anticipate malicious inference intentions in non-certified virtual sensors based on similarities to certified virtual sensors. We proposed a design based on three verification steps: two of them aim to anticipate inference intention and prevent malicious behaviors, and the other to prevent classified information release and inconsistent semantic perception. In order to provide privacy preservation and access authorization, we extend the concept of virtual sensors to express Access Control Models (ACMs) and Privacy-Preserving Data Mining Techniques (PPDMTs) that are techniques to verify release and usage authorization of personal information and to degrade *data utility* aiming to minimize classified information detection or detection accuracy.

7 AN EXPERIMENT ON PRIVACY-AWARE SENSING AS A SERVICE

CONTENTS

7.1 INTRODUCTION

In the previous chapter, we presented the Privacy-aware Virtual Sensor Model (PA-VSM), our privacy model for the IoT sensing defined through OWL representation and algorithms. This model is conceptualized based on *privacy-by-design* guidelines, providing a *privacy-by-policy* mechanism based on ontologies and semantic inference to evaluate policy conditions. The approach relies on the capacity to interpret the result that is produced on sensor data streams, anticipating privacy actions based on the semantic signature of KDDM processes that are provided along with the Sensing as a Service (S^2aaS).

The traditional S^2aaS paradigm (see Section 2.4.1) encompasses a variety of services that may include (raw, prepared, or processed) sensor data streams. However, the *semantic representation* of the data or information provided by these services is crucial to igniting the reasoning and inference employed in our privacy model. In this chapter, a privacy-aware S^2aaS is defined using the PA-VSM along with concepts of virtual sensor and Semantic Perception (SP). Additionally, in order to provide a testbed platform for evaluating the functional viability of this approach, an extension of a sensing platform architecture is presented for the proposed sensing service. Lastly, a use case is described to demonstrate how the sensor data stream, KDDM processes, PETs, and privacy policies are represented using the ontological framework defined in Section 6.2.

In the remain of this chapter, the privacy-aware S^2aaS for the IoT is formalized in Section 7.2, followed by its architecture in Section 7.3. Next, the use case is presented in Section 7.4. Lastly, in Section

7.5 results and conclusions are presented based on the preliminary results of the use case.

7.2 PRIVACY-AWARE SENSING AS A SERVICE

As discussed in Section 2.1.3, the Cloud service model provides a simpler interface for service customers and service providers. Through virtualization, Cloud service providers are able to bill only consumed services, while providing elasticity as demanded by Cloud customers. In the context of S^2aaS, Cloud native applications that collect, broadcast, process, and keep sensor data should incorporate this elasticity aspect to scale up their services according to some SLAs. The *in-network processing* paradigm of the S^2aaS, on which we ground our privacy strategy, benefits from the same elasticity and billing model. This service paradigm can be analogously extended to the concept of privacy-aware S^2aaS billing for *in-network privacy safe data processing*.

However, the conventional implementation of S^2aaS includes sensing of any types, which is not compatible with our proposed privacy model. The suitable service paradigm needs to incorporate data processing capacity for the sensor data streaming, enabling in-network KDDM and PET executions. Similar to the concept of Cloud of Things for S^2aaS, where edge computing platforms provide meaningful information from the in-network processing of sensing data [26], we envision a Cloud-IoT platform that intermediate *data consumers* and *data providers* outputting **only** *meaningful personal information which is not classified as private*. Thus, the virtualization of sensor in Cloud-IoT platform provides a network gateway where data collection, processing, and privacy preservation can be implemented together. This design guarantees that sensor data streams from private sensors are processed and then verified *on-the-fly*, creating a privacy preservation layer between *data providers* (*connected objects*) and *data consumers* (application and business layers). By shifting the conventional KDDM process execution toward the *joint-sphere*, several risks of privacy harm – as highlighted in Section 2.2 – can be addressed, such as anticipation of data processing, private data exchange, and minimization of released data quality.

Another requirement of the proposed privacy model is the *semantic representation* of the sensor data stream which can be achieved by semantic annotation of sensor data or by *semantic perception* as the output of KDDM processes. The former corresponds to the expressiveness of current ontologies used in S^2aaSs, while the latter is related to the type of KDDM processes that can be implemented. The semantic annotation of sensor data conventionally found in S^2aaSs [216, 60, 217, 218] is appropriate to provide *sensor discovery* and *sensor interoperability* techniques, but not to express personal information or KDDM processes used in the PA-VSM. This is due to the amount of accumulated processing, abstraction, information accuracy, and background knowledge that is incorporated during the KDDM execution and which can not be expressed in device-centric ontologies. The use of OPIS makes it possible to semantically represent aspects of these data processing scenarios and produced information, addressing the limitation of the SSN-O expressiveness. On top of that, the model defined in the semantic perception computing allows targeting KDDM processes that produce *semantic representation* from data streams, as required by the PA-VSM, such as DSM, SSR, and CEP approaches (Section 2.4.1).

The privacy enforcement is then achieved through incorporation of the PA-VSM into the sensing stream workflow of the virtual sensor, evaluating which Privacy-Enhancing Technology (PET) to apply preventively based on: (1) inference intention using the semantic perception output specification, (2) malicious inference intention using similarity measurement between certified and non-certified virtual sensors, and (3) inference results. Figure 7.1 depicts the Cloud-IoT platform incorporating our PA-VSM and the involved actors (data producer, data consumer, independent certifier, virtual sensor developer).

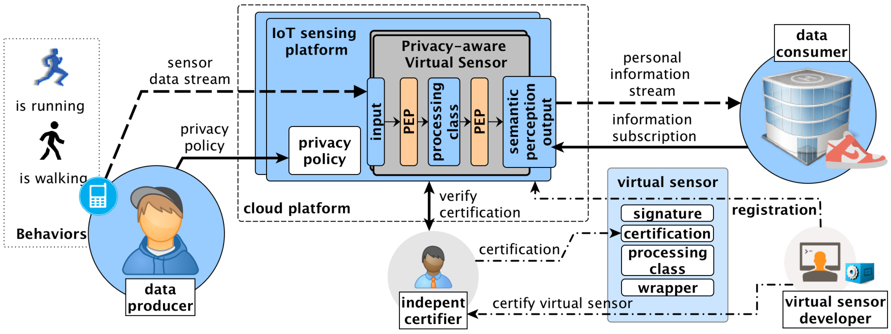

Figure 7.1 – Overview of our novel privacy-aware IoT sensing

The data producer registers her/his connected device, along with a privacy policy, that will be used by the sensing platform to deploy one or more *privacy-aware virtual sensors*. These virtual sensors must have the same input signature of the sensor data stream generated by his/her device and enforce privacy according to the provided policy. They intermediate data consumers and devices, receiving sensor data and streaming personal information. On the other side, data consumers must subscribe to the information streams that are available. In order to have the desired information available, privacy-aware virtual sensors must be developed. In Figure 7.1, virtual sensor developers and data consumers are depicted separately, since they can work independently. Still, similarly to complementary interest between end-users and application developers, they may influence each other in terms of need to produce and consume certain types of information. Another reason why virtual sensor developers are illustrated aside is the need of an optional certification process that envisions to shift the trust model toward independent parties that are not presumably interested in the value of information, focusing their business model in the certification or development processes instead of the personal information consume. While the virtual sensor developer plays an important role in the provision and consumption of information, the independent certifier meets the requirement of a third party that guarantees that virtual sensors behave according to their provided semantic signatures, improving the **reliability** and **viability** of the PA-VSM. This is achieved by minimizing the possibility of a malicious or erroneous virtual sensor identity (semantic signature) and increasing the number of possible virtual sensors that can be executed since PA-VSM accept the execution of non-certified virtual sensor by verifying its malicious intention using certified virtual sensors. In addition, virtual sensor developers can act as privacy protection agents, developing virtual sensors that implement PPDMTs and ACMs, which are not of interest of the data consumer. These virtual sensors that implement PETs need to be certified, because the privacy enforcement process relies heavily on them.

The personal information stream consists of a sequence of *semantic representation* based on OPIS The *semantic perception data type* class sets out a data structure specification, which can be used to express ontology axioms and the relationships between them (object properties) or data properties. This allows, for instance, associating *generalizations* and values, produced by data mining or machine learning techniques, to the *semantic representations*. As a consequence, virtual sensors can produce an output of complex type that may contain any type of data attach to its semantic representation. Since OPIS extends SSN-O and the evaluation engine of PA-VSM is modeled using Semantic Web technology, our proposed privacy-aware S^2aaS is compatible with available sensing platform that adopts SSN-O to annotate sensor data stream.

7.3 SENSING SERVICE ARCHITECTURE

Currently, many commercial and open-source IoT platforms are available, as surveyed in [219]. For the functional viability evaluation of our proposed privacy-aware S^2aaS, we selected the OpenIoT project to be extended as a testbed platform. OpenIoT is an open-source centralized IoT platform composed of open-source projects, including the eXtended Global Sensor Network (xGSN) middleware [27], which implements the concept of *virtual sensors*.

xGSN relies on semantic representations of sensor and observation metadata to implement the process of annotation and publishing sensor data on the Sensor Web. The system is constituted by the Global Sensor Network (GSN) middleware, the Linked Stream Middleware (LSM) and a quad-store database. GSN is a Virtual Sensor Network (VSN) that supports the rapid and simple deployment of a wide range of sensor network technologies [220]. It provides sensor virtualization that can be connected to several IoT networks protocols, such as CoAP, XMPP, MQTT etc. An initial set of protocols are bundled in the middleware, but new protocols can be implemented extending the *wrapper* class.

GSN provides data processing capabilities for virtual sensors through the concept of *processing class*, a self-containing Java class

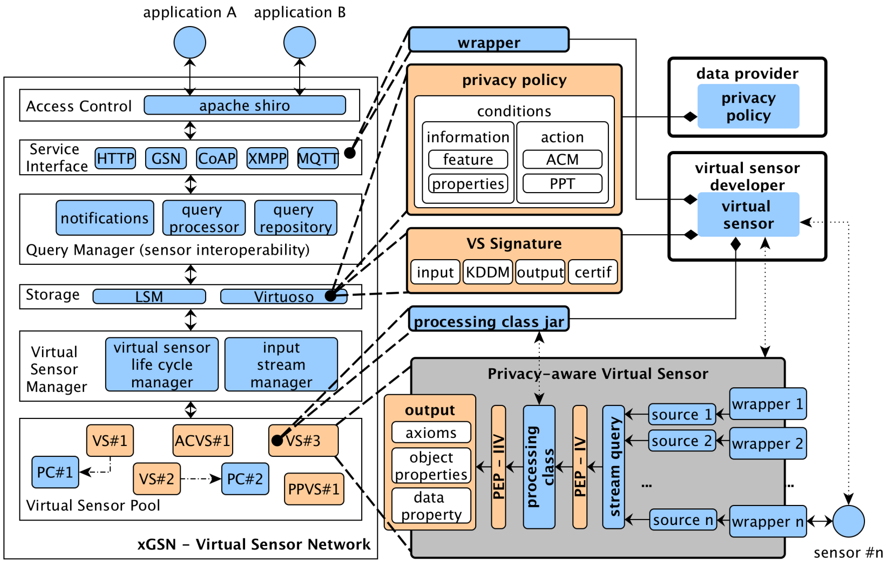

Figure 7.2 – Privacy-aware xGSN architecture. Normal arrows represents the workflow between xGSN layers. Diamond-end arrows represent part-hood relationship. Dotted arrows correspond association relationship. Orange boxes refer to the proposed extension.

that can be used to deliver data cleaning, preparation, and KDDM processing *on-the-fly*. LSM complements the GSN limitation to annotate sensor data using SSN-O to represent sensors and observations.

Virtuoso is the quad-store database in the OpenIoT project used to register semantic annotations for virtual sensors and observations. It provides a hybrid architecture for data access and integration, combining a Relational Database Management System (RDBMS) and Property Graph Data Management with SPARQL endpoint to deal with storage and retrieval of RDF-based quads (graph, subject, predicate, object). Therefore, SPARQL queries from our model can be executed directly over Virtuoso that contains sensor data, semantic signatures of virtual sensors – including PPDMTs and ACMs – and PPCs.

As depicted in Figure 7.2, we propose to implement our privacy model extending the xGSN architecture to : (1) represent virtual sensor meta-data (semantic signature) and privacy policies using OPIS; (2) specialize virtual sensor into a *privacy-aware virtual sensor* that implements our PA-VSM. The first goal is achieved by changing the template for virtual sensor signatures and internal java classes in the xGSN middleware to load the extended semantic signature. An XML file (VirtualSensorDescription.xml) containing the JAXB[1] binding schema defines a new format for the virtual sensor signature, so called *privacy-aware virtual sensor*. It is important to remark that original version of virtual sensor is compatible and can coexist with privacy-aware virtual sensors. The main different relies on the virtual sensor deployment. The privacy-aware virtual sensor must have the new semantic signature in order to be loaded correctly, otherwise, they will be de-

0. https://virtuoso.openlinksw.com/ (accessed on 26/04/2017)
1. http://www.oracle.com/technetwork/articles/javase/index-140168.html (accessed on 26/04/2017)

ployed without privacy protection. The JAXB binding schema defines the format expected of the virtual sensor signature. An example of a semantic signature for the virtual sensor that implements a "step predictor" algorithm is depicted in Listing 7.1.

Listing 7.1 – Virtual sensor signature file example

```
<semantic-perception-virtual-sensor name="sensor1" priority="10">
<certification> ... </certification>
<objective-specification class="..."/>
<processing-class>
<class-name>org.openiot.gsn.vsensor.StepPredictor</class-name>
<workflow>
<algorithm class="odm:OntoDM_368779">
algorithm 1
</algorithm>
<streams>
<stream name="input1">
<source alias="source1" sampling-rate="1" storage-size="1">
<address wrapper="cvs">...</address>
<query>
SELECT * FROM WRAPPER
</query>
</source>
<query>
SELECT * FROM source1
</query>
</stream>
</streams>
<output-structure>
<output-data-specification name="output1">
<semantic-perception-axioms>
<ontology-axiom iri="opis:behavioralAgent"
type="class">behavioral agent</ontology-axiom>
</semantic-perception-axioms>
</output-data-specification>
</output-structure>
<implementation id="weka.classifiers.trees.J48"> J48 tree from Weka
    toolkit version ...
</implementation>
<operators>
<operator-parameter name="prunning confidence" type="real"
    observation-value-region="dul:Amount">
10
</operator-parameter>
</operators>
</workflow>
</processing-class>
<addressing> <predicate>
<key>latitute</key><value> 92.2 </value>
<key>longitude</key><value> -22.2 </value>
</predicate> </addressing>
</semantic-perception-virtual-sensor>
```

The second goal is achieved by extending original Java classes of virtual sensors, which are aware of the new semantic signature and implement the algorithms defined in the PA-VSM. The main modification consists in extending the *AbstractVirtualSensor* Java class in two methods: *dataAvailable* and *dataProduced*. The *dataAvailable* method allows anticipating data delivery to the privacy-aware virtual sensor, where algorithms 2 and 3 are implemented. The *dataProduced* method is executed before outputting the privacy-aware virtual sensor result, where algorithm 4 is implemented. OPIS is made available along with the OpenIoT and loaded into Virtuoso quad-store, along with all its dependencies (imported ontologies). The code is

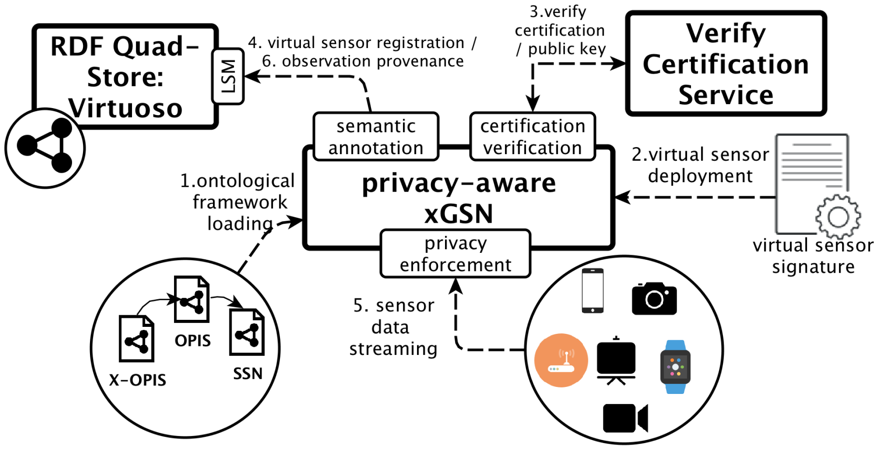

Figure 7.3 – xGSN instance and its dependencies

available in `https://github.com/thiagomoreirac/openiot.git` (accessed on 26/04/2017). In order to allow Virtuoso to correctly infer and retrieve information based on SPARQL queries, the OpenIoT virtual machine instance should be bundled with a base ontology, which, in our case it is formed by OPIS, SSN-O and an xOPIS that represents the ontological framework needed to represent sensor data, personal information and virtual sensors – as well as Privacy-Preserving Virtual Sensors (PPVSs) and Access Control Virtual Sensors (ACVSs) (see Section 6.2.4).

For each virtual sensor to be deployed, some files need to be injected in the OpenIoT virtual machine instance: (1) a virtual sensor signature file; (2) a *processing class* jar file, containing the Java class to be executed by the virtual sensor instance; (3) a wrapper jar file, containing the specialized wrapper class that implements some IoT network protocol (optional); and (4) a bulk loading RDF source file, containing any specific sub-domain ontology used by the virtual sensor (optional). Figure 7.3 illustrates these elements, their relationships, and processes.

After loading the ontological framework and injecting virtual sensor signatures, the certification verification is performed during the deployment of each virtual sensor, followed by its registration in the LSM (and consequently in the Virtuoso quad-store). As previously mentioned, the certification verification is optionally placed during the virtual sensor deployment, but other options can be implemented, such as including a periodic verification to guarantee that the certification is still valid and has not expired yet. The process to calculate similarity measurements, which are used during the verification of malicious inference intention, between the virtual sensor and the current deployed ones can also be executed during its deployment for performance matters. Since the semantic signature of a virtual sensor only changes during the installation, the similarity measurement is also a static value and can be safely stored. The similarity measurement is calculated using the SemMF library [2] [221]. We present

2. `http://semmf.ag-nbi.de` (accessed on 26/04/2017)

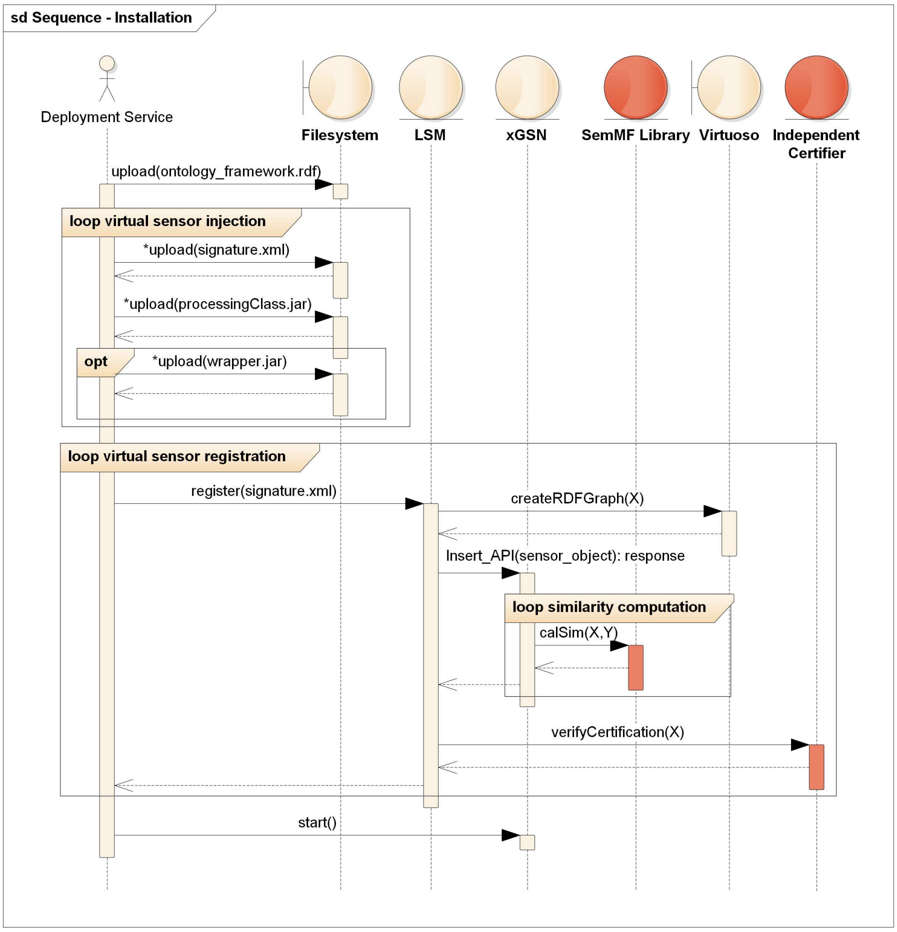

Figure 7.4 – Virtual sensor deployment sequence

the sequence diagram of OpenIoT instantiation in Figure 7.4. After the deployment of virtual sensors, the OpenIoT virtual machine instance is ready to be notified by connected devices about available sensor data, storing observations and related semantic annotations using the LSM.

The automatic deployment of the IoT sensing platform is an important part related to this work that would permit to evaluate performance and viability of the privacy-aware S^{2}aaS on-demand. A repository for virtual sensors and information about private devices (and their attached privacy policies) have to be made available so the deployment service could decide which virtual sensors need to be deployed and which devices could be connected in order to respond to an information subscription demand. Even though we understand its relevance and impact on the proposed privacy-aware S^{2}aaS, we refrain on focusing our contribution on the design of the sensing service architecture, which demonstrates the functional viability to implement this solution in a real IoT platform. A preliminarily study on the automation of the deployment service for privacy-aware sensor-clouds has been investigated in [222].

7.4 USE CASE

The objective of this use case is to demonstrate the viability of our ontology-based PA-VSM. In particular, the use case allows testing the instantiation of OPIS and SPARQL queries that are respectively the main semantic representation and the evaluation engine of our proposed privacy model. The first objective of this use case is to determine if the proposed ontology can be consistently instantiated, therefore, ready to be queried. By describing each step for representing personal information, PETs, classification taxonomies, and privacy policies, the usability of OPIS is evidenced by the verification of OWL Description Logic (OWLDL) language consistency and OPIS's competence to express all the elements proposed in our ontological framework.

Secondly, by describing and executing SPARQL queries used in the algorithms that constitute the PA-VSM, its viability is demonstrated, since their evaluation engine are basically queries executed against OWL inference engines. Apart from SPARQL queries, the complexity of these algorithms depends on the number of input and output data specifications, and the number of axioms and assertions outputted by the execution of virtual sensors. These numbers are empirically low, since a high number of inputs and outputs could be related to a not well-design KDDM process – in particular when considering IoT sensing service scenarios. On the other hand, the OWLDL language provides decidable computational properties for ontology consistency checking, class expression satisfiability, class expression subsumption checking, and instance checking. However, this can be classified as NP-Hard with a decidability up to a nondeterministic algorithm in time that is at most double exponential in the size of the input (2^{2^n}, for n is the size of the input) [111]. In other words, it can be unpractical to use it as an evaluation engine in the S^2aaS. For this reason, the second objective of this use case is evidencing the viability of the privacy model through the average response time of these SPARQL queries.

In order to illustrate, the example presented in Section 5.9.2 is extended, showing how the ontological framework based on OPIS can express personal information, PETs, classification taxonomies, and privacy policies. The portability from the conventional scenario of data processing described in algorithm 1 towards a data stream processing is based on the micro-batch execution model that transform a data stream into short batches, similarly to the model implemented by distributed computing approaches, such as Apache Spark. It is important to remark that the conversion from conventional to micro-batch execution models should be investigated for each case since streaming processing encompasses distinct parameters, configurations, and results from those achieved by batches and traditional execution models. However, in this use case, we focus on the investigation of the viability of our privacy model by reasoning about KDDM process, inputs, and output, instead of the accuracy of the KDDM process. Based

2. http://spark.apache.org/faq.html (accessed on 26/04/2017)

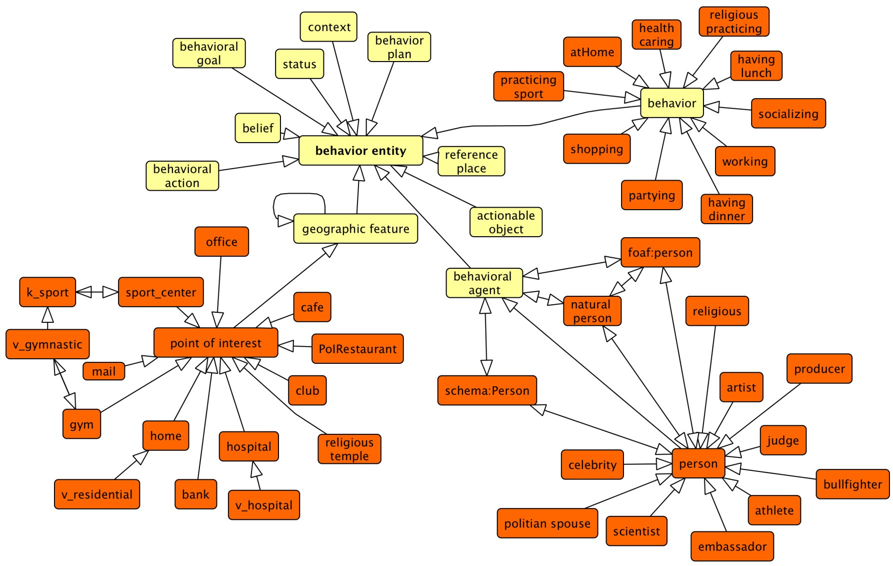

Figure 7.5 – Representation of Personal Information using OPIS. Yellow boxes represent OPIS classes. Orange boxes refers to imported classes.

on a predefined privacy policy that defines PPC in the form of *personal information classification* $\rightarrow$ *privacy enhancing technology*, PPDMTs and ACMs are executed preventively if a private personal information will/might be produced. This is possible because the semantic signature of the virtual sensor in execution contains the output specification using OPIS to describe the antecedent part of the condition (personal information) and the consequent part (a virtual sensor that implements a privacy enhancing technology that can be executed anticipatively). Since its result is a *semantic representation*, inference using OWL class subsumption and our proposed transversal classification (see Section 6.2.1) offer a powerful and formal verification to evaluate if an output is private.

In the next subsections, we describe the personal information that is used in this use case and how to represent it using the Personal Information Layer (PIL). Next, we introduce the virtual sensor that implements the KDDM process described in algorithm 1 as an SP in a data stream scenario, three virtual sensors that implement PETs and their semantic signatures using the Semantic Perception Layer (SPL). Next, the classification taxonomy used to support PPC definition is presented using the ontological framework provided in Section 6.2.3. Lastly, we present the SPARQL queries that constitute the core of our proposed PA-VSM, followed by the preliminary results. The definitions presented in this section are detailed in Annex 10 and available in `https://github.com/thiagomoreirac/opis/` (accessed on 26/04/2017).

7.4.1 Personal Information

OPIS provides the foundational structure to express personal information using the context of behavior. As explained in Section 5.7, two set of behavioral entities (vectors 5.1 and 5.2) cover the features involved during the creation or management of personal information. Analogously to the SSN-O, classes of OPIS should be extended by subsumption or by intention in order to be instantiated. This aspect of OPIS allows its adoption in different application domains, offering a base conceptual framework to classify information. Besides that, any of these classes can be extended to represent sub-domain concepts using existing domain ontologies. The alignment of OPIS with upper-level and middle-level ontologies makes it possible to import sub-domain ontologies that are available, such as those provided by the OBO Foundry, Ontology Design Patterns initiative [3] etc.

For this use case, we imported the DBPedia [4] and OSMonto [5]. The former provides a large-scale knowledge base extracted from Wikipedia that it is useful for being accessible worldwide and updated from the Wikipedia database. The latter provides geographic tags from the open-source project OpenStreetMaps [223] that associate spatiotemporal data to several types of information, such as point of interest, amenities, demographic data, and so forth. Figure 7.5 depicts a fragment of these concepts and their equivalence classes. For example, the behavioral entity 'geographic feature' is extended to describe a *point of interests* (PoI), and subsequently, specializations, such as PoIBank, PoICafe, PoIHome etc. These specializations are associated as equivalent (*owl:equivalentClass*) to categories of OSMOnto tags. As a consequence, a virtual sensor that intends to use the OpenStreetMaps APIs to access spatiotemporal information can concretely classify this data as personal through the 'geographic feature'.

In addition, the representation of persons using the DBPedia knowledge base is concretely expressed by associating the 'behavioral agent' concept to the class 'person' from DBPedia. Aiming to represent the scenario of human activities of [207] and to demonstrate some reasoning involving the *transversal classification*, the concept of 'behavior' is extended to be associated to point of interests through the object property *dul:hasLocation* (See Section 10.1 of Annex 10 for details). As a consequence, the behavior 'Practicing Sport' is associated with an object property 'is Location Of' several 'point of interests with sport tags, such as *v_gymnastics*, *v_soccer*, *v_golf*, *v_equestrian* etc.

SEMANTIC PERCEPTION DATATYPE The result expected from a *privacy-aware virtual sensor* is a *semantic representation* that may contain data values, which are traditionally outputted by KDDM processes. This *semantic representation* has a data type that needs to be interpreted by the privacy model in order to allow reasoning over *semantic observation values* that are produced by SPs. OPIS provides a data type specification for SP called *semantic perception data type* that restricts the KDDM result to be expressed in terms of *ontology axioms*,

3. http://ontologydesignpatterns.org/wiki/Ontology:Main (accessed on 26/04/2017)
4. http://wiki.dbpedia.org/Ontology (accessed on 26/04/2017)
5. http://wiki.openstreetmap.org/wiki/OSMonto (accessed on 26/04/2017)

object property assertions or *data property assertions*. *Ontology axioms* attribute corresponds to a set of axiom declarations that are used for the assertion in the other two sets.

It is important to notice that semantic representations can be concretely expressed both in TBox and ABox. As a consequence, a KDDM process can output an ontology axiom "Person" or an individual "Person A". The former is specified only using the *class axiom*, the latter is specified both using the *class axiom* and *class assertion* (known as individuals). Therefore, two conformity evaluations need to be performed before the PPC evaluation. Firstly, a *declaration evaluation* must be executed to verify if axioms are properly defined in the loaded ontology, checking if the declared *classes* are subclasses of *ssn:featureOfInterest*, *object properties* are subclasses of *opis:semanticProperties*, and *data properties* have subsumption of *ssn:property*. Secondly, an *assertion evaluation* verifies if class, object property, and data property assertions refer to these declared axioms. The loaded ontology is the conceptual framework built using PIL, as defined in this section, and available for the reasoning engine (in the moment of the privacy model execution). Table 7.1 presents the *axiom declarations* and *assertion set* that constitute a *semantic observation value*, along with their sets, and the evaluation for each set.

Attribute	Set	Set Evaluation
axiom declarations	$\mathcal{C}$: <classAxiom(C_1), ..., classAxiom(C_n)>	(C_i,IRI_i):hasIRI $[\![IRI_i]\!]_{\mathcal{O}} \sqsubseteq$ssn:FeatureOfInterest
	$\mathcal{P}$: <objectPropertyAxiom(OP_1), ... , objectPropertyAxiom(OP_n)>	(OP_i,IRI_i):hasIRI $[\![IRI_i]\!]_{\mathcal{O}} \sqsubseteq$ opis:semanticProperty
	$\mathcal{D}$: <dataPropertyAxiom(DP_1), ..., dataPropertyAxiom(DP_n)>	(DP_i,IRI_i):hasIRI $[\![IRI_i]\!]_{\mathcal{O}} \sqsubseteq$ssn:property
assertion set	<classAssertion(CA_1), ..., classAssertion(CA_n)>	(C_j,CA_i):hasPart $C_j \in \mathcal{C}$
	<objectPropertyAssertion(OPA_1), ..., objectPropertyAssertion(OPA_n)>	(OPA_i,C_i):hasPart (OPA_i,OP_i):hasPart (OPA_i,C_j):hasPart C_i,$C_j \in \mathcal{C}$ $OP_i \in \mathcal{P}$
	DataPropertyAssertion(<DPA_1>,...,<DPA_n>)	(DPA_i,C_i):hasPart (DPA_i,DP_i):hasPart (DPA_i,v_k):hasPart $C_i \in \mathcal{C}$ $DP_i \in \mathcal{D}$ ObservationValue(v_k)

Table 7.1 – Semantic Observation Value Instance Checking

According to the SSN-O paradigm for sensor observation, the abstraction region on which values are defined must be specified using the *observation value* class. This design permits to specify sensor data using any measurement systems or abstract region, such as the unit of measurement ontology [6], QUDT [7], QUOMOS [8], MUO [9] etc.

In the case of SP, the result is a *semantic representation*. However, neither RDF nor OWL is capable of expressing this relationship between an assertion of *observation value* class and a *semantic representation*. Technically, it is not possible to assert an object property between an individual (ABox) and an ontology axiom (TBox). OPIS addresses this issue by providing a *region* called *ontology axioms* – specialized in class axioms, object property axioms, and data property axiom – and a data property *hasIRI* that can be used to assert a data property between an individual of *ontology axiom* (observation value) and an IRI of an *ontology axiom* (semantic representation). For instance, in order to represent the class "Person" in the *semantic observation value*, a class axiom is asserted with a data property assertion *hasIRI* having a xsd:anyURI value as the IRI of class "Person". The *assertion evaluation* is possible because the *ontology axiom* has *axiom assertions*, which are used to assert object properties and data properties.

In order to provide the SSN-O compatibility, *semantic observation values* can refer to original sensor observation values using data property assertions. Since these values can be defined according to any *region*, OPIS provides the data type specification based on the OntoDT expressiveness (see Section 3.3.2.1) that allows application defining data format, as defined in the XML Schema for RDF [10]. In addition, OPIS design for axiom declaration and assertion permit to refer to data property and object property used in the *semantic observation value*, mapping them respectively to *ssn:property* and its extension *opis:semanticProperty*.

7.4.2 Semantic Signature

In this use case, one virtual sensor that implement the perception of human activity is described, along with three virtual sensors that implements the following PETs: Attribute-based Access Control (ABAC), Role-based Access Control (RBAC), and *k*-anonymity. Therefore, five semantic signatures are represented using SPL, presented in the next subsections. These signatures are available in Section 10.2 of Annex 10 and are loaded during the virtual sensor deployment (see Figure 7.4).

VIRTUAL SENSOR FOR HUMAN ACTIVITY PERCEPTION The semantic signature of the virtual sensor that implements human activity perception is composed of three main sections: (1) semantic perception process; (2) data types; and (3) algorithm implementations and pa-

6. https://bioportal.bioontology.org/ontologies/UO (accessed on 26/04/2017)
7. http://www.qudt.org/ (accessed on 26/04/2017)
8. https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=quomos (accessed on 26/04/2017)
9. http://idi.fundacionctic.org/muo/muo-vocab.html (accessed on 26/04/2017)
10. https://www.w3.org/TR/swbp-xsch-datatypes/ (accessed on 26/04/2017)

rameters. The semantic perception process contains key information about the sequence of algorithms, objectives for each step, and information about data type, observed property (and feature of interest), and the expected result in terms of Semantic Perception (SP). The virtual sensor has geographic location property specified as input in its first algorithm specification (*stopDetectionAlgorithm*) of the type *real*. In the output specification of the third algorithm specification (*calculateProbabilityPointOfInterestIAlgorithm*), the use of *ontology axiom* region is requested to represent the link between an individual of ontology axiom to a class definition. As explained previously, the limitation of OWL to express these types of relationship requires that we employ the *hasIRI* annotation property from an individual (ABox) to a class (TBox) by asserting the *class axiom*. For this virtual sensor output, three ontology axioms are defined: *point of interest class axiom*, *human activity class axiom*, and *'is location of' object property axiom*. The WGS84 data type represents the format of the geographic location coordinates that should be provided by the sensor, but also to compare between virtual sensors and PETs that have the same type of input and thus, compatible.

VIRTUAL SENSORS FOR ACCESS CONTROL MECHANISMS One of the main benefit of expressing PETs as virtual sensors using the concept of semantic signature is the direct implication of comparison and reasoning using OWL reasoners and SPARQL queries for all virtual sensors, including those that discovery personal information and those that preserve privacy. Queries used to identify *concurrent virtual sensors* (definition 14) can retrieve from the quad-store (Virtuoso) a list of virtual sensors that have in common a subset of input specification. In the case of virtual sensors that implement ACMs, they control personal information publishing, and, therefore, accept only *behavioral entity* as input/output in its data specification.

An Access Control Virtual Sensors (ACVSs) do not transform data stream, limiting their functionality to suppress data (attribute) if it is not allowed to be published. They have a specific semantic signature since they do not differ input and output data specification, as defined in Section 6.2.2. Figure 7.6 depicts the semantic signature of a ACVS that implements an ABAC technique. The most important information in this signature is the data specification that permits retrieve this virtual sensor to execute in the stream of any personal information since it refers to the *behavioral entity* class. Listing 7.2 provides the OWL Functional Syntax of this semantic signature. Another important issue in ACVS and PPVS is their certifications. Since they should be *trusted virtual sensors*, it is expected they are not provided by non-verified parties. The certification information is illustrated in the "..."^xsd:String data elements. Another ACVS is included in this use case to exemplify the diversity in PETs that can coexist, highlighting the *plurality* aspect of our approach. The Role-based Access Control (RBAC) technique is similarly represented with almost the same semantic signature. The decision of which technique will be executed depends on the PPC that is evaluated *on-the-fly*. The SPARQL

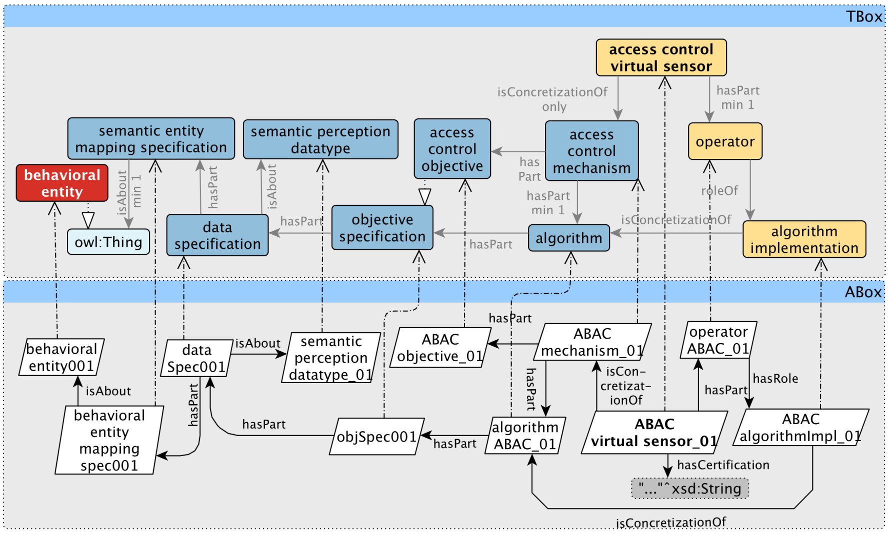

Figure 7.6 – TBox/Abox representation of an attribute-based access control virtual sensor. Dash-dotted lines represent instance-of (rdf:type). Dotted lines represent subsumption relationship. Red box, yellow boxes, blue boxes, and cyan box represent respectively PIL class, specification classes from SPL, implementation classes from SPL, and owl:Thing.

query should retrieve only the specified ACVS in the individual's policy.

Listing 7.2 – Fragment of semantic signature for an Attribute-Based Access Control (ABAC) tecnique using OPIS. Notation: OWL Functional Syntax.

```
ClassAssertion(<xopis:accessControlVirtualSensor> <ABACVirtualSensor
    _01>)
ClassAssertion(<obi:operator> <operatorABAC_01>)
ClassAssertion(<odm:algorithmImplementation> <ABACAlgorithmImpl_01>)
ClassAssertion(<xopis:accessControlMechanism> <ABACMechanism_01>)
ClassAssertion(<xopis:accessControlObjective> <ABACObjective_01>)
ClassAssertion(<iao:algorithm> <algorithmABAC_01>)
ClassAssertion(<iao:objectiveSpecification> <objSpec001>)
ClassAssertion(<iao:dataSpecification> <dataSpec001>)
ClassAssertion(<xopis:semanticPerceptionMappingSpecification> <
    behavioralEntityMappingSpec001>)
ClassAssertion(<opis:BehavioralEntity> <behavioralEntity1001>)

ObjectPropertyAssertion(<ro:has_part> <ABACVirtualSensor_01> <
    operatorABAC_01>)
ObjectPropertyAssertion(<ro:hasRole> <operatorABAC_01> <
    ABACAlgorithmImpl_01>)
ObjectPropertyAssertion(<obi:isConcretizationOf> <ABACVirtualSensor
    _01> <ABACMechanism_01>)
ObjectPropertyAssertion(<ro:has_part> <ABACMechanism_01> <
    ABACObjective_01>)
ObjectPropertyAssertion(<ro:has_part> <ABAClMechanism_01> <
    algorithmABAC_01>)
ObjectPropertyAssertion(<obi:isConcretizationOf> <ABACAlgorithmImpl
    _01> <algorithmABAC_01>)
ObjectPropertyAssertion(<ro:has_part> <algorithmABAC_01> <objSpec
    001>)
ObjectPropertyAssertion(<ro:has_part> <objSpec001> <dataSpec001>)
ObjectPropertyAssertion(<ro:has_part> <dataSpec001> <
    behavioralEntityMappingSpec001>)
ObjectPropertyAssertion(<iao:isAbout> <behavioralEntityMappingSpec
    001> <behavioralEntity001>)
```

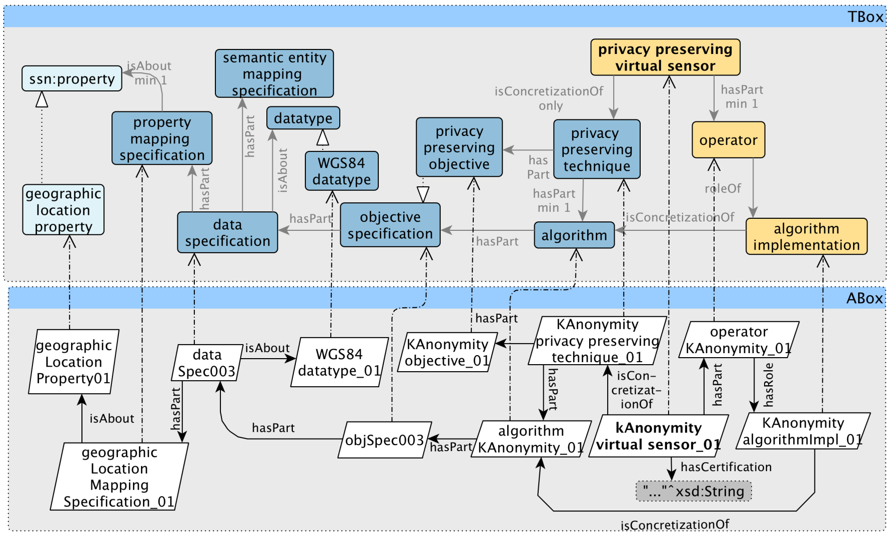

Figure 7.7 – TBox/Abox representation of a k-anonymity virtual sensor. Dash-dotted lines represent instance-of (rdf:type). Dotted lines represent subsumption relationship. Yellow boxes, blue boxes, and cyan box represent respectively specification classes from SPL, implementation classes from SPL, and SSN-O classes.

```
ObjectPropertyAssertion(<iao:isAbout> <dataSpec001> <
    semanticPerceptionDatatype001>)

AnnotationAssertion(<opis:hasCertification> <ABACVirtualSensor_01>
    "..."^xsd:String)
```

VIRTUAL SENSORS FOR DATA MINING PRIVACY PRESERVING TECHNIQUES The representation of PPVS is slightly different from ACVS's, while keeping the similarity to data specification (same for input and output). The difference is specifically in the data transformation process, which happens in the case of PPVS. The data stream sample is transformed into an anonymized data stream sample, using the *processing class jar* that bundles PPDMTs and parameters. Technically, PPVSs can have predefined configurations, but it also can retrieve preferences from a persistent file or table. In this use case, these preferences are considered static, but some possibilities to parameterized these PPDMT includes (but are not restricted to): parameters in PPC definitions (instead of simple pair of *personal information* → *privacy enhancing technology*), administrative interface for configuring persistent files, crowd-sourcing preferences based on web services, and so forth.

Since PPDMT are meant to decrease data utility without changing the format of the dataset, the semantic signature of PPVS should contain the observed property and data type aimed to have its data quality degraded. Figure 7.7 depicts a semantic signature of a PPVS that implements a *k*-anonymity technique. The *geographic location property* – extension of a *ssn:property* – is instantiated in order to be referred by the *property mapping specification*. Instead of just *semantic entity mapping specifications* and *semantic perception datatypes*, PPVS can have data specification of any data type and *property mapping specifications*. In the example of the *k*-anonymity virtual sensor, a WGS84 data type is used to define the sensor data observation from a geographic location sensor. Listing 7.3 presents a fragment of the semantic signature for a *k*-anonymity technique in OWL Functional syntax.

Listing 7.3 – Fragment of semantic signature for an k-anonymization technique using OPIS. Notation: OWL Functional.

```
ClassAssertion(<xopis:accessControlVirtualSensor> <
    kAnonymityVirtualSensor_01>)
ClassAssertion(<obi:operator> <operatorKAnonymity_01>)
ClassAssertion(<odm:algorithmImplementation> <KAnonymityAlgorithmImpl
    _01>)
ClassAssertion(<xopis:privacyPreservingTechnique> <
    KAnonymityPrivacyPreservingTechnique_01>)
ClassAssertion(<xopis:privacyPreservingObjective> <
    KAnonymityObjective_01>)
ClassAssertion(iao:algorithm <algorithmKAnonymity_001>)
ClassAssertion(iao:objectiveSpecification <objSpec003>)
ClassAssertion(<odm:odm:dataSpecification> <dataSpec003>)
ClassAssertion(<wgs84:WGS84Datatype> <WGS84Datatype_01>)
ClassAssertion(<opis:propertyMappingSpecification> <
    geographicLocationMappingSpecification_01>)
ClassAssertion(<xopis:geographicLocationProperty> <
    geographicLocationProperty01>)

ObjectPropertyAssertion(obi:isConcretizationOf <
    kAnonymityVirtualSensor_01> <KAnonymityPrivacyPreservingTechnique
    _01>)
ObjectPropertyAssertion(ro:has_part <kAnonymityVirtualSensor_01> <
    operatorKAnonymity_01>)
ObjectPropertyAssertion(ro:hasRole <operatorKAnonymity_01> <
    KAnonymityAlgorithmImpl_01>)
ObjectPropertyAssertion(ro:has_part <
    KAnonymityPrivacyPreservingTechnique_01> <KAnonymityObjective
    _01>)
ObjectPropertyAssertion(ro:has_part <
    KAnonymityPrivacyPreservingTechnique_01> <algorithmKAnonymity
    _001>)
ObjectPropertyAssertion(obi:isConcretizationOf <
    KAnonymityAlgorithmImpl_01> <algorithmKAnonymity_001>)
ObjectPropertyAssertion(ro:has_part <algorithmKAnonymity_001> <
    objSpec003>)
ObjectPropertyAssertion(ro:has_part <objSpec003> <dataSpec003>)
ObjectPropertyAssertion(iao:isAbout <dataSpec003> <wgs84:WGS84
    Datatype_01>)
ObjectPropertyAssertion(ro:has_part <dataSpec003> <
    geographicLocationMappingSpecification_01>)
ObjectPropertyAssertion(<iao:isAbout> <
    geographicLocationMappingSpecification_01> <
    geographicLocationProperty_01>)

AnnotationAssertion(opis:hasCertification <kAnonymityVirtualSensor
    _01> "..."\~{}xsd:String)
```

7.4.3 Classification Taxonomy

The use of a classification taxonomy is part of our privacy model to facilitate the reference of personal information in concepts related to an application domain or easier for end-users to classify their own information. The information classification structure proposed in Section 6.2.1 is based on DUL ontology, using the classes *concept* that can be used to define a classification taxonomy. As depicted in Figure 7.8, we classify the behavioral entities according to eight life contexts:

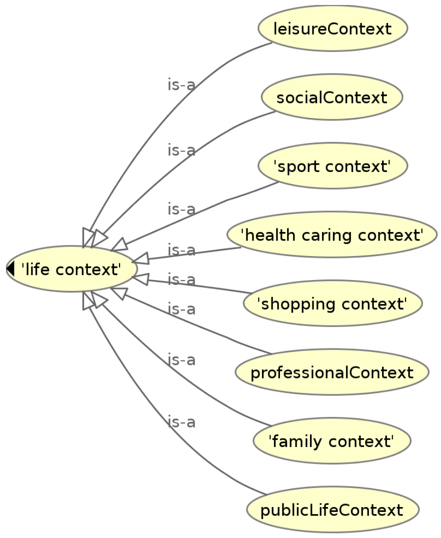

Figure 7.8 – Classification taxonomy example

leisure context, social context, sport context, health caring context, shopping context, professional context, family context, and public life context.

The classification taxonomy should be defined along with the object property assertions of *classifies* between the instance of a specific *concept* and an individual of a classified personal information (defined in PIL) *ObjectPropertyAssertion(ssn:classifies <healthRelatedContext_01> <v_hospital_01>)*. Instead of defining rules for every user, the classification taxonomy allows reusing a set of classifications by multiple individuals. In addition, the transversal classification enables an efficient classification that goes beyond the traditional direct classification, using the inference power of OWL reasoning engines. For instance, let health caring context be used to define a PPC of the type *health caring context* $\rightarrow$ *k-anonymity*. The data stream can be anonymized even if the result – human activity – is not directly classified. This is possible because another *trusted virtual sensor* can use the same sensor data sample and discovery points of interest tagged with *v_hospital*, which are related to a *health-related context*, and therefore, capable of being retrieved using the PPC about health-related contexts.

It is important to remark that the PPC and classification taxonomies use individuals (class instances) to assert object properties that allows reasoning over the set of instances and facts. However, the SPARQL queries defined in the PA-VSM are defined in a way to consider the classes of those instances, instead of specific individuals, to perform its transversal classification inference. In this use case, we associate each life context to its respective human activities (behaviors) as presented in Table 7.2.

life context	behavioral entity
leisure context	PracticingSport / Partying / having Dinner / havingLunch
social context	havingDinner / Partying / ReligiousPracticing / havingLunch
sport context	PracticingSport
health caring context	HealthCaring
shopping context	Shopping
professional context	Working
family context	AtHome

Table 7.2 – Mapping between life contexts and behavioral entities

7.4.4 Privacy Policy

Privacy Policy Conditions (PPCs) are the concrete representation of individual's preference for her privacy. It is also expressed in OWL assertions that associate classification, PET, and optional a time interval. In our use case, as depicted in Listing 7.4, we define one PPC that associates 'sport context' to *k*-anonymity virtual sensor (kAnonymiyVitualSensor_01), and 'social context' to the attribute-based access control virtual sensor (ABACVirtualSensor_01).

Listing 7.4 – Examples of privacy policy conditions. Notation: OWL Functional.

```
ClassAssertion(<privacyPreservingCondition> <
      privacyPreservingCondition_01>)
ObjectPropertyAssertion(<dul:isSettingFor> <
      privacyPreservingCondition_01> <copis:sportContext001>)
ObjectPropertyAssertion(<includesPrivacyPreservingVirtualSensor> <
      privacyPreservingCondition_01> <kAnonymityVirtualSensor_01>
ClassAssertion(<AccessControlCondition> <accessControlCondition_02>)
ObjectPropertyAssertion(<dul:isSettingFor> <accessControlCondition
      _02> <copis:socialContext001>))
ObjectPropertyAssertion(<includesPrivacyPreservingVirtualSensor> <
      accessControlCondition_02> <ABACVirtualSensor_01>
```

7.4.5 SPARQL Queries

PA-VSM has three main PEPs implemented in three algorithms: the inference intention verification, the malicious inference intention verification, and the inference verification. In the Algorithm 2, two SPARQL queries are executed: **RetrieveACVSForVS** and **RetrievePPVSForVS**. As defined in the previous chapter, **RetrieveACVSForVS** sets for every specified output in a given virtual sensor, the transversal classification verifies if there is any classification defined in the privacy policy of the user to an ACVS. We assembled these concepts described in Tables 6.1 , 6.5, and 6.7 into one query presented in Listing 7.5.

The query is divided into sections and has translated OBO codes, such as IAO_000136, into correspondent label references, for readability matters. The current virtual sensor's ID (:ID) is identified in the FILTER clause, and an ontology axiom (class, object property, or data property) should be returned, along with the ACVS, if a PPC links one of the virtual sensor output to a classified ontology axiom. The transversal and direct classification translate the path between a direct classified behavioral entity to a classification concept (defined in the classification taxonomy). Then, the virtual sensor output specification is extracted using its semantic signature. At last, if the virtual sensor ontology axiom output is a subclass of the classified behavioral entity, one ACVS is returned.

Listing 7.5 – RetrieveACVSForVS SPARQL query.

```
PREFIX dul: <http://www.loa-cnr.it/ontologies/DUL.owl#>
PREFIX opis: <https://.../thiagomoreirac/opis/master/opis.owl#>
PREFIX ppol: <https://...gomoreirac/opis/master/privacy_policy.owl#>
PREFIX obi: <http://purl.obolibrary.org/obo/>
PREFIX ro: <http://www.obofoundry.org/ro/ro.owl#>
PREFIX iao: <http://purl.obolibrary.org/obo/>
PREFIX odm: <http://kt.ijs.si/panovp/OntoDM#>
PREFIX odt: <http://kt.ijs.si/panovp/OntoDT#>
PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
SELECT  ?iVirtualSensor ?cOntologyAxiom ?iACVS WHERE {{
#=== Privacy Policy Condition =======
?privacyPolicyCondition ppol:includesAccessControlVirtualSensor ?
    iACVS.
?privacyPolicyCondition dul:isSettingFor ?iDirectClassification.
?privacyPolicyCondition a ?PPC. ?PPC rdfs:subClassOf+ ppol:
    PrivacyPolicyCondition.
#=== Transversal Classification ====
?iDirectClassification a/rdfs:subClassOf* ?cTransversalClassification
    .
?cTransversalClassification rdfs:subClassOf* dul:Concept; ^a  ?
    iTransversalClassification.
#=== Direct Classification =======
?iTransversalClassification dul:classifies ?cTCOntologyAxiom.
?cTCOntologyAxiom a/rdfs:subClassOf+ opis:OntologyAxiom; opis:hasIRI
    ?cTransversalClassifiedAxiom.
?cTransversalClassifiedAxiom (rdfs:subClassOf|owl:equivalentClass|^
    owl:equivalentClass)+ opis:BehavioralEntity.
?cTransversalClassifiedAxiom ((rdfs:subClassOf/(owl:someValuesFrom|
    owl:allValuesFrom|owl:onClass))|((owl:unionOf/rdf:rest*/rdf:first
    )*))* ?cClassifiedAxiom.
?iClassifiedAxiom a/(rdfs:subClassOf|^owl:equivalentClass|owl:
    equivalentClass)* ?cClassifiedAxiom.
#=== Output VS ===============
?iVirtualSensor a/rdfs:subClassOf* opis:VirtualSensor.
?iVirtualSensor obi:obi:isConcretizationOf ?iSPP. ?iSPP a/rdfs:
    subClassOf* opis:SemanticPerceptionProcess.
?iSPP ro:has_part|dul:hasPart ?iAlgorithm. ?iAlgorithm a/rdfs:
    subClassOf iao:iao:algorithm.
?iAlgorithm ro:has_part|dul:hasPart ?iObjectiveSpec. ?iObjectiveSpec
    a/rdfs:subClassOf* iao::objectiveSpecification.
?iObjectiveSpec ro:has_part|dul:hasPart ?iOutputDataSpec. ?
    iOutputDataSpec a/rdfs:subClassOf* odm:odm:dataSpecification.
?iOutputDataSpec ro:has_part|dul:hasPart ?iMapSpec. ?iMapSpec a/rdfs:
    subClassOf* odm:odm:mappingSpecification.
?iOutputDataSpec iao:iao:isAbout ?iDatatype. ?iDatatype a/rdfs:
    subClassOf* odt:datatype.
?iMapSpec iao:iao:isAbout ?iOntologyAxiom.  ?cOntologyAxiom rdfs:
    subClassOf* ?IRIType.
#=== RetrieveACVSForVS ==============
?cOntologyAxiom rdfs:subClassOf*/^a ?iClassifiedAxiom.
} FILTER (?IRIType IN (ssn:FeatureOfInterest, ssn:Property))
FILTER (?iVirtualSensor = :ID)} LIMIT 1
```

Similarly, **RetrievePPVSForVS** is defined based on PPC which includes PPVS as part of the policy condition. We present its SPARQL query in Listing 7.6, highlighting the differences in comparison to **Re-**

trieveACVSForVS. The complete version of this query can be found in Listing 10.3 in Appendix 10 The query retrieves PPVSs that are set in the user's PPCs that has a path to an ontology axiom transversally classified of the given virtual sensor (:ID). Based on these PPVSs, we select those whose input specification is a subset of the given virtual sensor input specification.

Listing 7.6 – RetrievePPVSForVS SPARQL query.

```
PREFIX ...
SELECT  ?iPPVS ?cOntologyAxiom ?cProperty ?cFeatureOfInterest ?
    iDatatype WHERE {{
#=== Privacy Policy Condition =======
?privacyPolicyCondition ppol:includesPrivacyPreservingVirtualSensor ?
    iPPVS.
...
#=== Transversal Classification ====
...
#=== Concrete Classification =======
...
#=== Output VS ===============
?iVirtualSensor ... opis:OntologyAxiom. ?cOntologyAxiom rdfs:
    subClassOf* ?IRIType.
#=== Input VS  ===============
?iVirtualSensor a/rdfs:subClassOf* opis:VirtualSensor.
?iVirtualSensor obi:isConcretizationOf ?iSPP. ?iSPP a/rdfs:subClassOf
    * opis:SemanticPerceptionProcess.
?iSPP ro:has_part|dul:hasPart ?iAlgorithm. ?iAlgorithm a/rdfs:
    subClassOf iao:algorithm.
?iAlgorithm ro:has_part|dul:hasPart ?iObjectiveSpec. ?iObjectiveSpec
    a/rdfs:subClassOf* iao:objectiveSpecification.
?iObjectiveSpec ro:has_part|dul:hasPart ?iInputSpec. ?iInputSpec a/
    rdfs:subClassOf* odm:descriptiveDataSpecification.
?iInputSpec ro:has_part|dul:hasPart ?iMapSpec. ?iMapSpec a/rdfs:
    subClassOf* odm:mappingSpecification.
?iInputSpec iao:iao:isAbout ?iDatatype. ?iDatatype a/rdfs:subClassOf*
     odt:datatype.
?iMapSpec iao:iao:isAbout|(iao:iao:isAbout/opis:hasIRI) ?cProperty. ?
    cProperty rdfs:subClassOf* ssn:Property.
OPTIONAL {?cProperty ssn:isPropertyOf ?cFeatureOfInterest. ?
    cFeatureOfInterest rdfs:subClassOf* ssn:FeatureOfInterest.}
#=== Input PPPVS ===============
?iPPVS a/rdfs:subClassOf* opis:VirtualSensor.
?iPPVS obi:isConcretizationOf ?iSPP2. ?iSPP2 a/rdfs:subClassOf* opis:
    SemanticPerceptionProcess.
?iSPP2 ro:has_part|dul:hasPart ?iAlgorithm2. ?iAlgorithm2 a/rdfs:
    subClassOf iao:algorithm.
?iAlgorithm2 ro:has_part|dul:hasPart ?iObjectiveSpec2. ?
    iObjectiveSpec2 a/rdfs:subClassOf* iao:objectiveSpecification.
?iObjectiveSpec2 ro:has_part|dul:hasPart ?iInputSpec2. ?iInputSpec2 a
    /rdfs:subClassOf* odm:descriptiveDataSpecification.
?iInputSpec2 ro:has_part|dul:hasPart ?iMapSpec2. ?iMapSpec2 a/rdfs:
    subClassOf* odm:mappingSpecification.
?iInputSpec2 iao:iao:isAbout ?iDatatype. ?iDatatype a/rdfs:subClassOf
    * odt:datatype.
?iMapSpec2 iao:iao:isAbout  ?cProperty. ?cProperty rdfs:subClassOf*
    ssn:Property.
OPTIONAL {?cProperty ssn:isPropertyOf ?cFeatureOfInterest. ?
    cFeatureOfInterest rdfs:subClassOf* ssn:FeatureOfInterest.}
#=== RetrievePPVSForVS ===============
?cOntologyAxiom (rdfs:subClassOf*|^rdfs:subClassOf*)/^a ?
    iClassifiedAxiom.
}
FILTER (?IRIType IN (ssn:FeatureOfInterest, ssn:Property))
FILTER (?iVirtualSensor != ?iPPVS)
FILTER (?iVirtualSensor = :ID)
} LIMIT :LIMIT
```

In Algorithm 3, the SPARQL query **SimilarConcurrentCertifiedVS** is built using the concept of virtual sensors that share an input specification subset but diverge in the result. The rationale behind this relies on the premise of malicious inference intention for non-certified virtual sensors. The SPARQL query defined to express that is presented in Listing 7.7, and encompasses the formal model defined in Tables 6.9 and 6.10.

Listing 7.7 – RetrievePPVSForVS SPARQL query.

```
PREFIX opis: <https://.../thiagomoreirac/opis/master/opis.owl#>
PREFIX ppol: <https://...gomoreirac/opis/master/privacy_policy.owl#>
PREFIX obi: <http://purl.obolibrary.org/obo/>
PREFIX ro: <http://www.obofoundry.org/ro/ro.owl#>
PREFIX iao: <http://purl.obolibrary.org/obo/>
PREFIX odm: <http://kt.ijs.si/panovp/OntoDM#>
PREFIX odt: <http://kt.ijs.si/panovp/OntoDT#>
PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
SELECT  ?iVirtualSensor ?iConVS WHERE {{

#=== ConVS ===============
SELECT  ?cProperty2 ?cFeatureOfInterest2 ?iDatatype2 WHERE {
?iConVS a/rdfs:subClassOf* opis:VirtualSensor.
?iConVS opis:hasCertification ?certification.
?iConVS ro:has_part|dul:hasPart ?iVSSim2. ?iVSSim2 obi:
     isSpecifiedOutputOf/a obi:similarityCalculation.
?iConVS obi:isConcretizationOf ?iSPP2. ?iSPP2 a/rdfs:subClassOf* opis
     :SemanticPerceptionProcess.
?iSPP2 ro:has_part|dul:hasPart ?iAlgorithm2. ?iAlgorithm2 a/rdfs:
     subClassOf iao:algorithm.
?iAlgorithm2 ro:has_part|dul:hasPart ?iObjectiveSpec2. ?
     iObjectiveSpec2 a/rdfs:subClassOf* iao:objectiveSpecification.
?iObjectiveSpec2 ro:has_part|dul:hasPart ?iInputSpec2. ?iInputSpec2 a
     /rdfs:subClassOf* odm:descriptiveDataSpecification.
?iInputSpec2 ro:has_part|dul:hasPart ?iMapSpec2. ?iMapSpec2 a/rdfs:
     subClassOf* odm:mappingSpecification.
?iInputSpec2 iao:isAbout ?iDatatype2. ?iDatatype2 a/rdfs:subClassOf*
     odt:datatype.
?iMapSpec2 iao:isAbout  ?cProperty2. ?cProperty2 rdfs:subClassOf* ssn
     :Property.
OPTIONAL {?cProperty2 ssn:isPropertyOf ?cFeatureOfInterest2. ?
     cFeatureOfInterest2 rdfs:subClassOf* ssn:FeatureOfInterest.}.
FILTER NOT EXISTS {
SELECT ?cProperty ?cFeatureOfInterest ?iDatatype WHERE {
#=== VS  ===============
?iVirtualSensor a/rdfs:subClassOf* opis:VirtualSensor.
?iVirtualSensor ro:has_part|dul:hasPart ?iVSSim. ?iVSSim obi:
     isSpecifiedOutputOf/a obi:similarityCalculation.
?iVirtualSensor obi:isConcretizationOf ?iSPP. ?iSPP a/rdfs:subClassOf
     * opis:SemanticPerceptionProcess.
?iSPP ro:has_part|dul:hasPart ?iAlgorithm. ?iAlgorithm a/rdfs:
     subClassOf iao:algorithm.
?iAlgorithm ro:has_part|dul:hasPart ?iObjectiveSpec. ?iObjectiveSpec
     a/rdfs:subClassOf* iao:objectiveSpecification.
?iObjectiveSpec ro:has_part|dul:hasPart ?iInputSpec. ?iInputSpec a/
     rdfs:subClassOf* odm:descriptiveDataSpecification.
?iInputSpec ro:has_part|dul:hasPart ?iMapSpec. ?iMapSpec a/rdfs:
     subClassOf* odm:mappingSpecification.
?iInputSpec iao:isAbout ?iDatatype. ?iDatatype a/rdfs:subClassOf* odt
     :datatype.
?iMapSpec iao:isAbout  ?cProperty. ?cProperty rdfs:subClassOf* ssn:
     Property.
OPTIONAL {?cProperty ssn:isPropertyOf ?cFeatureOfInterest. ?
     cFeatureOfInterest rdfs:subClassOf* ssn:FeatureOfInterest.}.
FILTER (?cProperty=?cProperty2 && ?cFeatureOfInterest=?
     cFeatureOfInterest2 && ?iDatatype=?iDatatype2 && ?iVSSim=?iVSSim2
     )
FILTER (?iVirtualSensor != ?iConVS) }}
```

```
#=== SimilarConcurrentCertifiedVS ===============
?iVSSim2 ?pDataProperty  ?dSimilarity. ?pDataProperty rdfs:
    subPropertyOf owl:dataProperty.
FILTER (?dSimilarity >= :threshold)
} }}
```

At last, in Algorithm 4 the inference of virtual sensors is verified in terms of format (data type) and content. For each result produced by a given virtual sensor, the SPARQL query **RetrieveACVSForAxiom** is executed to retrieve an ACVS for each classified ontology axiom. The rationale is similar to the **RetrieveACVSForVs**, except that we retrieve ACVS directly from axiom definition (:ID) instead of virtual sensor output specification. Listing 7.8 presents the **RetrieveACVSForAxiom** query.

Listing 7.8 – RetrievePPVSForAxiom SPARQL query.

```
PREFIX dul: <http://www.loa-cnr.it/ontologies/DUL.owl#>
PREFIX opis: <https://.../thiagomoreirac/opis/master/opis.owl#>
PREFIX ppol: <https://...gomoreirac/opis/master/privacy_policy.owl#>
PREFIX obi: <http://purl.obolibrary.org/obo/>
PREFIX ro: <http://www.obofoundry.org/ro/ro.owl#>
PREFIX iao: <http://purl.obolibrary.org/obo/>
PREFIX odm: <http://kt.ijs.si/panovp/OntoDM#>
PREFIX odt: <http://kt.ijs.si/panovp/OntoDT#>
PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
SELECT  ?cOntologyAxiom ?iACVS WHERE {{
#=== Privacy Policy Condition =======
?privacyPolicyCondition ppol:includesAccessControlVirtualSensor ?
    iACVS.
?privacyPolicyCondition dul:isSettingFor ?iDirectClassification.
?privacyPolicyCondition a ?PPC. ?PPC rdfs:subClassOf+ ppol:
    PrivacyPolicyCondition.
#=== Transversal Classification ====
?iDirectClassification a/rdfs:subClassOf* ?cTransversalClassification
    .
?cTransversalClassification rdfs:subClassOf* dul:Concept; ^a  ?
    iTransversalClassification.
#=== Direct Classification =======
?iTransversalClassification dul:classifies ?cTCOntologyAxiom.
?cTCOntologyAxiom a/rdfs:subClassOf+ ?cTransversalClassifiedAxiom.;
    rdfs:subClassOf opis:BehavioralEntity.
?cTransversalClassifiedAxiom ((rdfs:subClassOf/(owl:someValuesFrom|
    owl:allValuesFrom|owl:onClass))|((owl:unionOf/rdf:rest*/rdf:first
    )*))* ?cClassifiedAxiom.
?iClassifiedAxiom a/(rdfs:subClassOf|^owl:equivalentClass|owl:
    equivalentClass)* ?cClassifiedAxiom.
} FILTER (?cOntologyAxiom = :ID) } LIMIT 1
```

7.4.6 Results

The queries presented in the previous subsection were executed using a virtual machine in Oracle VirtualBox 5 [11], Lubuntu 14.04 64bit, 5GB of RAM, duo-core, with hardware virtualization (VT-x), in a hosting machine with 16GB of RAM, Intel® Core™ i7-5500 (Broadwell GT2), Ubuntu 16.04 64bit. The image provided by the OpenIoT project were used to deploy the environment, which contains the Virtuoso 6 as a back-end for SPARQL endpoint.

11. https://www.virtualbox.org/ (accessed on 26/04/2017)

SPARQL Query	average response time
RetrieveACVSForVS	6 seconds
RetrievePPVSForVS	33 seconds
SimilarConcurrentCertifiedVS	21 seconds
RetrieveACVSForAxiom	2 seconds

Table 7.3 – Preliminar results in executing PA-VSM SPARQL queries

The first result is related to the consistency checking of the set of ontology axioms and assertions used in the use case. As demonstrated in this section, the semantic representation of the entities of the ontological framework for this use case could be instantiated, while keeping the consistency of the ontology axioms and facts. The second result is related to the response time of the SPARQL queries that were observed to measure their impact on the privacy model. The preliminary results evidence that all queries returned some result (no time-out), and consequently have a complexity that can be handled by current SPARQL endpoint solutions. In general, this result can be extended from Virtuoso to other solutions, since Virtuoso performance is well ranked [224]. The maturity of SPARQL endpoints is vital for our strategy based mostly on ontology representation and semantic technology. Table 7.3 presents the approximately average response time of this preliminary experiment.

The response time is directly related to the size of the RDF pattern graphs used to request the result set, as reported in some investigations [225, 226, 227]. In our use case, the smallest and fastest SPARQL query **RetrieveACVSForAxiom** contains 7 RDF triple patterns, 6 RDF path patterns, and has an average response time of 2 seconds. While the biggest and slowest contains 14 RDF triple patterns, 49 RDF path patterns, and has an average response time of 33 seconds.

A supplementary analysis comes in need to investigate how to improve these average response time. Approaches for query cache, for instance, can address part of the SPARQL response time by caching results for identical queries in static RDF graphs [228, 229]. This meets our privacy model requirements, once the results of those queries achieve a constant if no new virtual sensor, classification taxonomy, PET, and PPC is installed in the system. In a real S^2aaS use case, the first requests for query execution by privacy-aware virtual sensors will perform with an initial footprint that will be minimized once most of the situations of sensing are known by the query caching system. Still using the cache system solution, an improvement of the average time can be reached by triggering an automatic caching during the installation of new PETs and virtual sensors, or after an update operation in the user's privacy policy conditions and classification taxonomy.

We remark that, despite the fact that SPARQL-DL [210] were employed to facilitate the explanation of our model in chapter 6, in practice, this technology is not mature enough to support this level of complexity and requests, returning time-out results and instability. In fact, few implementations of this SPARQL endpoint is available and most are out-dated.

In terms of result set regarding this use case, the algorithms returned the following outputs for the virtual sensor <:HumanActivityPerceptionVS001> and the virtual sensor <:practicingSport>:

- **RetrieveACVSForVS**: No result, because the virtual sensor output is defined as pointOfInterest and behavioralAgent class axioms and the 'has location of' object property axiom. Since the two PPCs classify 'sport context' and 'social context', no ACVS should be applied. It is important to remark that, in these case, it is possible that this virtual sensor produces either classified or non-classified point of interest, that's why no access control should be executed before the next PEPs.
- **RetrievePPVSForVS**: results a set of (<KAnonymityVirtualSensor_01>, <geographicLocationProperty>, - (no feature defined), <odt:realDatatype>). There is a transversal classification between 'sport context' to pointOfInterest (`(sport context, classifies, practicing sport), (practicing sport, has location in, PoISportCenter), (PoISportCenter, equivalent, k_sport), (k_sport, sub class of, point of interest) )`. Since there is the possibility that a classified behavioral entity is extracted from the input data stream, the PPVS k-anonymity should be executed.
- **RetrieveACVSForAxiom**: in the case of virtual sensor execution with a *k-anonymized* sensor data stream sample results in 'practicing sport', the result of this SPARQL query is `(<:roleBasedAccessControlVirtualSensor_01>)`. There is a direct classification between 'sport context' and 'practicing sport'. Therefore, in this case, the roleBasedAccessControlVirtualSensor_01 will be executed and according to the mechanism, this information will be released or not.

Another investigation with more set of virtual sensors, using the concept of certification must be executed in order to evaluate how the concept of similarity signature will impact the efficiency of the privacy preservation strategy and the overall performance. For this reason, the execution of the query **SimilarConcurrentCertifiedVS** could not be evaluated in the scope of this use case.

7.5 CONCLUSION

In this chapter, we have defined a novel *privacy-aware* Sensing as a Service (S^2aaS) based on our novel Privacy-aware Virtual Sensor Model (PA-VSM) and our Ontology for Personal Information on the Sensor Web (OPIS) extending the architecture of a real IoT platform. The rationale used to design the sensing services using enabling technologies that are available in the Cloud-IoT were described. The architecture of the service, based on the xGSN architecture, is presented, along with its novel components and the new sequence of virtual sensor installation and deployment.

The usability of the OPIS was also demonstrated by the instantiation of personal information, PETs, classification taxonomy, and PPC described in the use case. The viability of our proposed PA-VSM was evidenced by executing SPARQL queries that constitute the evaluation engine of our privacy model during the execution of virtual sensors in the xGSN. For this matter, Virtuoso was used as the main OWL reasoner and SPARQL endpoint to analyze queries and their response time. The solution employed OPIS as the ontological framework to describe virtual sensors, personal information, and PPCs, which demonstrate the viability of our model. Preliminary results were achieved evidencing usability and viability of our approach.

The efficiency of our approach relies on the flexibility and performance of Semantic Web technology in S^2aaS of available platforms. We also remark that OWL represents the most adopted format to represent knowledge and semantics currently. Therefore, our solution benefits from future improvement in its performance and, as the information systems gradually adopted the OWL representation and the Linked Data model, more information will be available to be mapped into our ontological framework that models and classifies personal information.

The innovative way of considering an observed feature of interest as personal information differs from traditional *Personal Information Management* systems. However, as the IoT connects not just physical sensor and actuators, but also information systems, we believe that the exchange of this information will converge, so provenance could be provided in the interconnected world.

8 CONCLUSIONS AND PERSPECTIVES

The aim of this chapter is to reiterate the main purpose and contributions, as well as, layout directions that emerge as extensions to the work hereby presented.

8.1 SUMMARY AND KEY CONTRIBUTIONS

In this work, we investigated the idea that personal information defined in behavioral contexts on the Sensor Web could leverage mechanisms of personal privacy preservation in the IoT. We derived our vision for personal privacy on the IoT-based on modern privacy engineering principles and the assessment of privacy harm factors (see Section 2.2). Our privacy paradigm was inspired by the idea of reusing enabling technologies and available privacy preservation solutions to deliver a user-friendly, plural and efficient privacy mechanism. For this, we reviewed the IoT technology, architectures, and middlewares to identify the best way to interfere with the IoT sensing to enforce *privacy by design*. Moreover, we aimed to deliver a *privacy by policy* mechanism developed on top of the IoT enabling technologies that would offer a more cognitive and contextual interface to the IoT user.

From this vision, we identified three main challenges. Firstly, the limitation of current ontologies to represent personal information and contextual information on the Sensor Web that would guarantee compatibility to the existing IoT sensing services. Secondly, the trust model based on the *consumer side* that kept the execution of KDDM processes on the *recipient sphere*, requiring sensor data to be sent to an environment where privacy adversaries could exploit breaches and extract private information from sensor data stream. Thirdly, the incompatibility of Privacy-Enhancing Technologies (PETs) to a plurality privacy paradigm that would allow performing the adequate privacy-preserving technique according to the individual's privacy policy, informational and personal contexts.

For the first challenge, we investigate how the informational context and situations in the real world could be used to represented personal information. The current *de facto* standard for sensor observation in the majority of IoT platforms, the SSN-O, is too technical to support the classification of personal information for the final user. For this matter, in Chapter 5, we proposed OPIS, an ontology for personal information on the Sensor Web that extends the concepts of SSN-O. OPIS has two layers that could be used individually: Personal Information Layer (PIL) and Semantic Perception Layer (SPL). Each of these layers corresponds to contributions in the state of the art of

Semantic Sensor Network (SSN). The PIL was developed using the concepts of Behavior Computing (BC) (see Section 3.4), extending the concepts of features of interest that participate during the IoT sensing in a behavioral context. By defining a set of interrelated classes, called *behavioral entity classes* (see Section 5.7), we proposed a base structure to classify any type of personal information that could be captured by the IoT sensing. *Behavioral entity classes* should be specialized in a real use cases, but its structure provides a web of features that are related to the observed individual. Additionally, the mapping of these *behavioral entity classes* to BFO-based ontologies and DUL contributes to the integration of sensor data to applications of different domains, such as environmental science, general medical science, neurobehaviors, mental functioning and so forth.

In the SPL, our contributions rely on the conceptualization of *virtual sensors* (see Section 5.8). We addressed the limitation of SSN-O by incorporating the KDDM representation from the OntoDM to represent *virtual sensors*. Similarly to the SSO paradigm of SSN-O that provides the semantic representation of sensor and its output, we defined the Semantic Perception (SP) paradigm to *virtual sensors*. In the SP paradigm, *virtual sensors* produce features of interest from observed properties of other features of interest. This contribution leverages the representation of sensor output on the Sensor Web, guaranteeing semantic representation for each output. Since *virtual sensors* are defined as data processing units, its semantic representation was proposed using OntoDM entities to specify its KDDM process and implementation. The *virtual sensors* output was represented using classes of OntoDM execution layer, such as dataset and data type. The direct contribution of this semantic representation of *virtual sensor* is the data provenance that covers aspects that SSN-O does not.

For the second challenge, we propose to shift the data processing of sensor data towards the Cloud-IoT infrastructure where PEP could prevent the execution of unwanted or malicious KDDM processes. Based on the data analytics approaches for data processing (see Section 2.3), such as CEP and DSM, and the capacity of the IoT-Cloud elasticity, we propose a *privacy by design* model in Chapter 6, called Privacy-aware Virtual Sensor Model (PA-VSM). Our model encapsulates the data processing using a two-fold PEP that executes verification before and after the data processing step. The PEP is constituted by three verification steps and uses Semantic Web technology to define and evaluate privacy policies (see Section 6.3.2). Our ontology OPIS is then used as the foundation to define an expressive ontological framework to specify privacy policy conditions. This ontological framework is formally defined to provide a transversal classification for interrelated personal information, employing the concept of transitive from the OWL expressiveness. This novel ontological framework innovates by proposing in one single structure the association of personal information, classification structure, and Privacy-Enhancing Technology (PET), which permit multiple classifications dynamically associated to multiple PEPs. The evaluation of privacy policy conditions in our model is practically implemented using SPARQL queries. The main advantage of expressing sensor data, personal information, privacy policy conditions, virtual sensors, and privacy-preserving techniques using the same ontological foundation is the possibility to query and produce personal information in the same semantic platform and to associate virtual sensors and privacy mechanisms dynamically. The PA-VSM is designed to evaluate if the information generated by *virtual sensors* is authorized according to the individual's privacy policy. In addition, our model uses the semantic representation of the KDDM process, so-called *semantic signature*, to anticipate inference intention and prevent the unnecessary execution of *virtual sensors* that would produce information that is not authorized to be released. Moreover, a supplementary step is conceived in PA-VSM to detect malicious inference intention based on the concept of virtual sensor certification and semantic signature similarity. This inference anticipation and detection of malicious activities also configures an innovation for ACM systems.

For the last challenge, we propose to extend the concept of *virtual sensor*, which has processing capabilities, to implement PETs. Similar to *virtual sensors*, Privacy-Preserving Virtual Sensors (PPVSs) and Access Control Virtual Sensors (ACVSs) can be represented in terms of its inputs, outputs, and KDDM process. As a consequence, PPVS can be executed *by demand* according to the individual's privacy policy and the sensor data stream. Additionally, PPVS and ACVS can be developed and deployed in the same way *virtual sensors* are. By implementing and representing PETs at the same level of KDDM, we were able to innovate in the way privacy is provided as service. Besides that, the preventive step for malicious inference detection, that uses the power of trusted KDDM techniques, leverages the performance of *privacy by policy* mechanism that can use the same type of technology, employed by privacy adversaries, to enforce privacy.

Our *privacy by design* model and *privacy by policy* mechanism meet the three privacy engineering principles of *plurality*, *contextuality*, and *contestability*. Besides that, they address the privacy harm factors observed by Ohm [25], specifically the detection and prevention of unintended and malicious data processing techniques, the trust model, the minimization of data release, and the accountability provided by the data provenance related to *virtual sensors*. Still marginally, our approach allows that motives are declared using the certification mechanism and *semantic signature* metadata.

As a further contribution, in chapter 7, we proposed to implement our *privacy by design* model and *privacy by policy* mechanism using a real IoT platform: the xGSN. In order to test the viability and performance of our approach, the implementation of a privacy-aware Sensing as a Service (S^2aaS) based on the PA-VSM demonstrated satisfiable results and performance. The second result of this implementation is the suitability of OPIS to represent personal information and support sensing services in a real IoT platform. The privacy mechanism implemented in an S^2aaS corresponds to a contribution in real IoT sensing scenarios.

Our contributions comprehensively address the research questions presented in Section 1.4. The PA-VSM efficiently provides a *privacy by design* model that prevents malicious data processing to gain access to personal information or private sensor data, while controlling access to personal information classified as private. The ontological framework that was developed to support the *privacy by policy* mechanism is grounded on OPIS, and therefore, on the behavioral context of the IoT user. At last, our implementation of a privacy-aware S^2aaS addresses the last research question.

8.2 FUTURE WORK

The in-depth analysis of the implementation of our approach in the privacy-aware S^2aaS served to explicit concerns we had about the performance of our solution. Since our *privacy by policy* mechanism depends mostly on SPARQL queries and semantic reasoning, we observed a number of challenges and future research that we classify in terms of each chapter of our contribution.

OPIS In order to develop our ontology for personal information, we employ the OWLDL expressiveness for representing classes and relationship between them. In our experiment, we noticed the impact of this expressiveness in terms of reasoning processing time. Further investigation on how to express these classes and properties using OWL profiles could be handy in making semantic reasoning faster. Concerning its complete compatibility with OBO Foundry guidelines, OPIS does not follow best practices for class and property naming; and even though it is available in a public repository, it is not available in the central directory of OBO for ontologies.

The SSN-O is an *on-going* project. Recently, a working group to improve and evolve SSN-O has been updating its structure to be more modular and to incorporate recent versions of the DUL ontology. This is an important step for semantic sensor network and should be incorporated by OPIS in the future.

The concept of sensor provenance is leveraged with the OPIS semantics. However, a study to investigate the compatibility with PROV-O [230] would contribute to integrate the virtual sensor provenance of OPIS into systems that adopt PROV-O. Still related to the concept of provenance, the automatic discovery of semantic perception process should be investigated. Currently, tools for extracting sequence diagrams of compiled Java programs are available, which makes possible to map theses sequences diagrams to semantic perception process using OPIS. This would automatize part of the process of certification. Similarly to what happens in the research domain of Meta-Mining (MM), higher-level reasoning and even data mining techniques could help to discover patterns in the mapped semantic perception process related to malicious inference.

PA-VSM Our *privacy by design* model is an initial step toward the definition of holistic privacy mechanisms for IoT sensing that takes into consideration an ontological framework to defines its functionalities. Even though we formally define this ontological framework independently of our OPIS, both were created interchangeably influenced by each other. Further investigation on how our PA-VSM behaves with other ontologies, such as pure OntoDM and domain specific ontologies for information classification (other than personal information), should be investigated.

Works to minimize the footprint performance generated by the PA-VSM privacy enforcement can be pursued, such as SPARQL query caching by using RDBMS established features, neural systems to improve the efficiency of querying about virtual sensors and recurrent results produced by similar SPPs.

The data flow proposed in the PEPs of PA-VSM could be leverage to the current standards for access control systems, such as delegations and permission expiration described in the XACML.

Lastly, we believe that a *graphical user interface* would also be of great support to guide the process of definition classification structures, define semantic signatures of virtual sensors, PPVSs, and ACVSs.

Part III

APPENDIX

9 ONTODM: AN ONTOLOGY FOR DATA MINING

9.1 INTRODUCTION

In this Appendix, we describe the ontological framework of OntoDM that is composed by several ontology imports. The design of our ontology OPIS reflects OntoDM structure since both follow ontology engineering best practices and because many concepts in OPIS are extended from the OntoDM.

9.2 ONTOLOGICAL FRAMEWORK

In the upper level, as presented in Figure 9.1, BFO provides an *ontological framework* that classifies entities into two basic classes: *continuants* (or *substantial entities*) and *occurents* (or *processual entities*). *Continuants* refer to entities that exist fully at any time (or at least, during a wide temporal interval perspective, for example, the human lifespan) in which it exists completely and its existence is not dependable on time, such as KDDM techniques, *spatial region* or a person. *Continuants* can be classified in *independent continuants* and *dependent continuants*. The former refers to bearer of *quality* or a *realizable entities*, such as a physical object (*material entity*), *object boundary*, or a *site*. The latter consists of entities that depend on one or multiple *independent continuants*. *Dependent continuants* can be specialized in *generically dependent continuant* or *specifically dependent continuant*. The former represents an entity that depends on another entity to exists. For instance, a file, software or an implementation of a KDDM that needs (any) a computer to exist. The latter represents classes that require some specific instances of independent continuant, such as the role of a researcher or the function of a data aggregation algorithm. Conversely

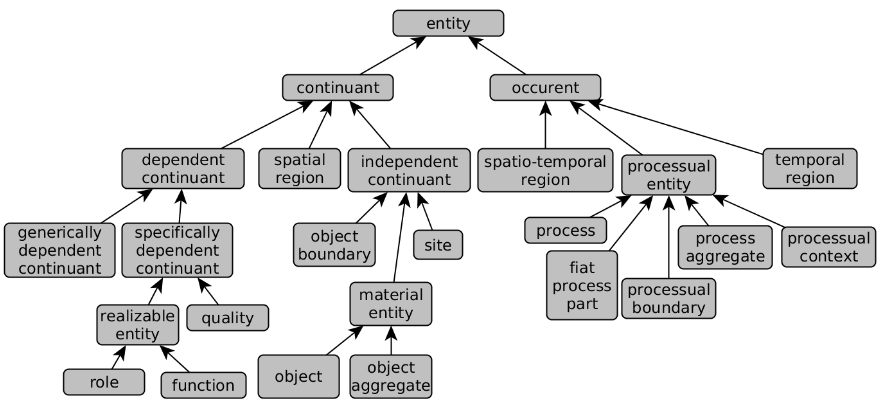

Figure 9.1 – Fragment of the BFO. Arrows represent subclass axiom

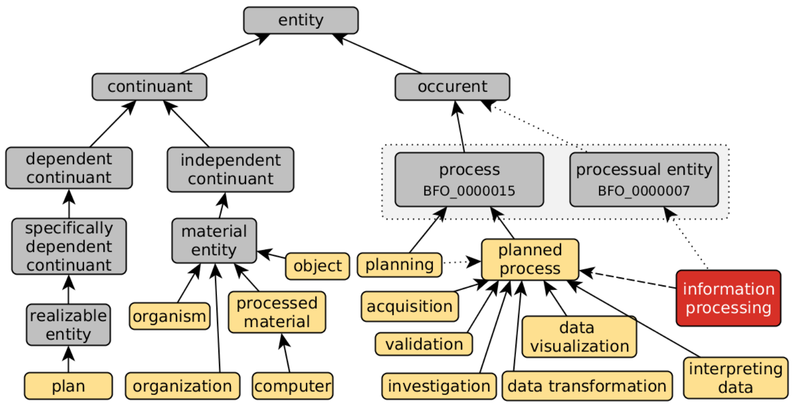

Figure 9.2 – Fragment of BFO, OBI and SWO. Gray boxes represent BFO entities. Yellow boxes represent OBI entities. Red box represents SWO entity. Arrows represent subclass axiom.

to *continuants*, *occurents* are temporal constrained, happening or being unfolded through a specific interval, such as the execution of a KDDM process. *Temporal regions*, *spatiotemporal regions* and *processual entities* are specializations of *occurents*. The *processual entity* exists in time while being executed or happening and depends on some entity. Microorganism lifespan, KDDM process, behavior, task are some example of *processual entities*.

All three middle-level ontologies incorporated in OntoDM (OBI, IAO, and SWO) are based on BFO. OBI was originally defined to represent scientific investigations, specialized in specifying the biological object of research and phases of the scientific experimentation. A fragment of OBI and SWO classes, extending BFO entities, are presented in Figure 9.2. *Plans* are specified based on BFO *realizable entity*, which is realized in a *planned process*. OBI specializes BFO *material entity* as physical entities, such as *organism*, *organizations* (*research organizations*), *processed material*, *computers*, *object*, and so on. OBI also described a set of *processual entities* that realize *plans*, such as *planning*, *acquisition*, *validation*, *investigation*, *data transformation*, *data visualization*, *interpreting data*, and *information processing*.

Originally in OntoDM, SWO *information processing* extends the BFO concept of *processual entity* (BFO_0000007) because of its alignment with the BFO version 1.1, that represents *process* and *processual entity* separately. In its current version, the process class BFO_0000007 was discarded and replaced by the BFO_0000015 which now is called *process*. In the remainder of this thesis, we assume *information processing* as a subclass of *planned process*, and we correct the import of SWO that pointed to the obsolete class *processual entity* (BFO_0000007), instead of the new BFO process class (BFO_0000015). This issue is evidenced in Figure 9.2 through dotted arrows and it is addressed in order to minimize confusion and reasoning problems in our approach.

A *processual entity* realizes a *plan* according to a *plan specification* pursuing one or more *objective specifications*. OntoDM ability to describe informational entities, processes that specifically produce and consume information, and realizations of those information entities

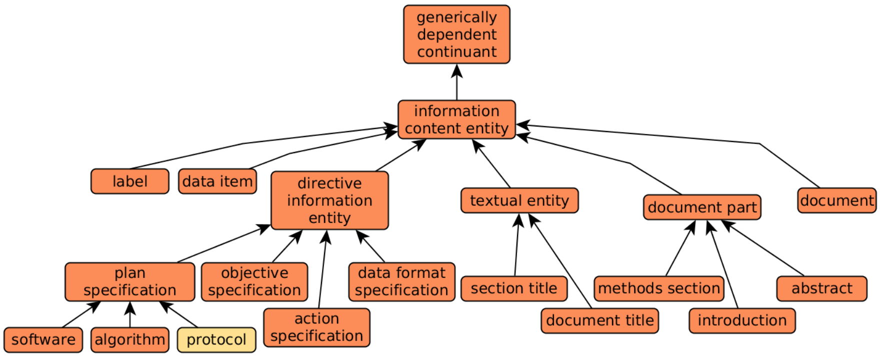

Figure 9.3 – Fragment of IAO. Orange boxes represent IAO classes. Yellow box represents OBI classes. Arrows represent subclass axiom.

are represented using IAO and OBI. Figure 9.3 illustrates a fragment of the IAO and OBI extension. ICE denote any information entity that describes another entity, such as *data item*, *label*, *textual entity*, *document*, *document part*, figures, etc. A *data item* and a *label*, for instance, intend to represent, respectively, a datum about some entity and a description that explains some *data item*. The DIE is an important information content entity that refers to BFO *realizable entities*. *Data format specification*, *action specification*, *objective specification*, and *plan specification* are specialized from this class. The *data format specification* is defined as an *information content entity* that describes how another *information content entity* should be encoded. The *action specification* is defined as some action that is realizable by some agent. The *objective specification* is defined as a process goal. A *plan specification* includes one or several *objective specifications* and *action specifications*. When realized (executed), a *plan specification* intends to achieve objectives following specified actions. *Algorithms*, *software*, *programming languages*, and *protocols* are specializations of *plan specification*. *Textual entities*, *documents*, and *document parts* are classes that describe concretizations of KDDMs or structured inputs and documentation related to them. IAO allows breaking down these *textual entities*, enabling information acquisition from documents (and texts), as well as information presentation.

The set of OntoDM relations is based on the BFO version 2.0, OBO RO, IAO, OBI, Experiment ACTions Ontology (EXACT)[1], and LAboratory Ontology for Robot Scientists (LABORS). EXACT ontology contains representation for experiment actions and it can be used for the full formalization of protocols of bio-medical investigations [231]. LABORS extends Exposé and imports OBO RO as a set of relations to represent scientific experiments in a form that allows computational agents to reason autonomously about hypothesis formation, experiment planning, and analysis of results. Table 9.1 presents OntoDM relations, their origins, and inverse relations.

1. http://www.aber.ac.uk/en/cs/research/cb/dss/exact/ (accessed on 26/04/2017)

Origin	Relation	Inverse Relation
BFO 2.0 & OBO RO	hasPart	partOf
	hasParticipant	participatesIn
	hasActiveParticipant	isActiveParticipantOf
	preceedes	preceededBy
	inheresIn	bearerOf
	hasQuality	isQualityOf
	hasRole	isRoleOf
	isConcretizedAs	isConcretizationOf
	realizes	isRealizedBy
IAO	isAbout	
	denotes	
	qualityIsSpecified-as	isQualitySpecificationOf
OBI	achievesPlannedObjective	objectiveAchievedBy
	hasSpecifiedInput	isSpecifiedInputOf
	hasSpecifiedOutput	isSpecifiedOutputOf
	isManufacturedBy	
EXACT	hasInformation	
LABORS	hasRepresentation	isRepresentationOf

Table 9.1 – OntoDM relations

10 USE CASE: RDF FILES

10.1 PERSONAL INFORMATION

```
EquivalentClasses(<dbpedia:Person> <opis:BehavioralAgent>)
EquivalentClasses(<envo:geographicFeature> <opis:GeographicFeature>)
EquivalentClasses(<osmonto:can_have_k_atm> <xopis:PoIBank>)
EquivalentClasses(<osmonto:can_have_k_cuisine> <xopis:PoIRestaurant>)
EquivalentClasses(<osmonto:k_religion> <xopis:PoIReligiousTemple>)
EquivalentClasses(<osmonto:k_shop> <xopis:PoIMall>)
EquivalentClasses(<osmonto:k_sport> <xopis:PoISportCenter>)
EquivalentClasses(<osmonto:v_cafe> <xopis:PoICafe>)
EquivalentClasses(<osmonto:v_gymnastics> <xopis:PoIGym>)
EquivalentClasses(<osmonto:v_hospital> <xopis:PoIHospital>)
EquivalentClasses(<osmonto:v_nightclub> <xopis:PoIClub>)
EquivalentClasses(<osmonto:v_pharmacy> <xopis:PoIPharmacy>)
EquivalentClasses(<osmonto:v_residential> <xopis:PoIHome>)
SubClassOf(<xopis:PointOfInterest> <opis:GeographicFeature>)
SubClassOf(<xopis:AtHome> <opis:Behavior>)
SubClassOf(<xopis:AtHome> ObjectSomeValuesFrom(<dul:hasLocation> <
    xopis:PoIHome>))
SubClassOf(<xopis:HealthCaring> <opis:Behavior>)
SubClassOf(<xopis:HealthCaring> ObjectSomeValuesFrom(<dul:hasLocation
    > <xopis:PoIHospital>))
SubClassOf(<xopis:HealthCaring> ObjectSomeValuesFrom(<dul:hasLocation
    > <xopis:PoIPharmacy>))
SubClassOf(<xopis:Partying> <opis:Behavior>)
SubClassOf(<xopis:Partying> ObjectSomeValuesFrom(<dul:hasLocation> <
    xopis:PoIClub>))
SubClassOf(<xopis:PoIBank> <xopis:PointOfInterest>)
SubClassOf(<xopis:PoICafe> <xopis:PointOfInterest>)
SubClassOf(<xopis:PoIClub> <xopis:PointOfInterest>)
SubClassOf(<xopis:PoIGym> <xopis:PointOfInterest>)
SubClassOf(<xopis:PoIHome> <xopis:PointOfInterest>)
SubClassOf(<xopis:PoIHospital> <xopis:PointOfInterest>)
SubClassOf(<xopis:PoIMall> <xopis:PointOfInterest>)
SubClassOf(<xopis:PoIOffice> <xopis:PointOfInterest>)
SubClassOf(<xopis:PoIPharmacy> <xopis:PointOfInterest>)
SubClassOf(<xopis:PoIReligiousTemple> <xopis:PointOfInterest>)
SubClassOf(<xopis:PoIRestaurant> <xopis:PointOfInterest>)
SubClassOf(<xopis:PoISportCenter> <xopis:PointOfInterest>)
SubClassOf(<xopis:PracticingSport> <opis:Behavior>)
SubClassOf(<xopis:PracticingSport> ObjectSomeValuesFrom(<dul:
    hasLocation> <xopis:PoISportCenter>))
SubClassOf(<xopis:ReligiousPracticing> <opis:Behavior>)
SubClassOf(<xopis:ReligiousPracticing> ObjectSomeValuesFrom(<dul:
    hasLocation> <xopis:PoIReligiousTemple>))
SubClassOf(<xopis:Shopping> <opis:Behavior>)
SubClassOf(<xopis:Shopping> ObjectSomeValuesFrom(<dul:hasLocation> <
    xopis:PoIMall>))
SubClassOf(<xopis:Socializing> <opis:Behavior>)
SubClassOf(<xopis:Socializing> ObjectSomeValuesFrom(<dul:hasLocation>
     <xopis:PoICafe>))
SubClassOf(<xopis:Working> <opis:Behavior>)
SubClassOf(<xopis:Working> ObjectSomeValuesFrom(<dul:hasLocation> <
    xopis:PoIOffice>))
SubClassOf(<xopis:havingDinner> <opis:Behavior>)
SubClassOf(<xopis:havingDinner> ObjectSomeValuesFrom(<dul:hasLocation
    > <xopis:PoIRestaurant>))
SubClassOf(<xopis:havingLunch> <opis:Behavior>)
```

```
SubClassOf(<xopis:havingLunch> ObjectSomeValuesFrom(<dul:hasLocation>
      <xopis:PoIRestaurant>))
```

10.2 SEMANTIC SIGNATURE OF HUMAN ACTIVITY PERCEPTION VIRTUAL SENSOR

```
ClassAssertion(<opis:IRIDatatype> <opis:IRIDatatype001>)
ObjectPropertyAssertion(<iao:isAbout> <opis:IRIDatatype001> <ssvs:
      behavioralAgent_classAxiom>)
ObjectPropertyAssertion(<iao:isAbout> <opis:IRIDatatype001> <ssvs:
      humanActivity_classAxiom>)
ObjectPropertyAssertion(<iao:isAbout> <opis:IRIDatatype001> <ssvs:
      isLocationOf_objectPropertyAxiom>)
ObjectPropertyAssertion(<iao:isAbout> <opis:IRIDatatype001> <ssvs:
      pointOfInterest_classAxiom>)
ClassAssertion(owl:DataProperty <opis:hasIRI>)
ClassAssertion(<dul:Agent> <ssvs:Agent007>)
ObjectPropertyAssertion(<ssn:hasProperty> <ssvs:Agent007> <ssvs:Agent
      007Geolocation>)
ClassAssertion(<x-opis:geographicLocationProperty> <ssvs:Agent007
      Geolocation>)
ClassAssertion(<odm:outputDataSpecification> <ssvs:
      HumanActivityOutputSpecification001>)
ObjectPropertyAssertion(<iao:isAbout> <ssvs:
      HumanActivityOutputSpecification001> <opis:
      semanticPerceptionDatatype001>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:
      HumanActivityOutputSpecification001> <odm:outputSpecification>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:
      HumanActivityOutputSpecification001> <ssvs:
      humanActivityMappingSpecification001>)
ClassAssertion(<opis:VirtualSensor> <ssvs:HumanActivityPerceptionVS
      001>)
ObjectPropertyAssertion(<obi:isConcretizationOf> <ssvs:
      HumanActivityPerceptionVS001> <ssvs:
      humanActivityPerceptionProcess001>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:HumanActivityPerceptionVS
      001> <ssvs:calculateProbabilityPointOfInterestOperator001>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:HumanActivityPerceptionVS
      001> <ssvs:selectPointOfInterestOperator001>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:HumanActivityPerceptionVS
      001> <ssvs:stopDetectionOperator001>)
ClassAssertion(<odm:parameter> <ssvs:SpatialAccuracy001>)
ClassAssertion(<odm:parameterSetting> <ssvs:
      SpatialAccuracyPoICalculation001>)
ObjectPropertyAssertion(<iao:isQualitySpecificationOf> <ssvs:
      SpatialAccuracyPoICalculation001> <ssvs:SpatialAccuracy001>)
ClassAssertion(<odm:parameterSetting> <ssvs:
      SpatialAccuracyPoISelection001>)
ObjectPropertyAssertion(<iao:isQualitySpecificationOf> <ssvs:
      SpatialAccuracyPoISelection001> <ssvs:SpatialAccuracy001>)
ClassAssertion(<odt:realFieldComponent> <ssvs:WGPS84
      LatitudeFieldComponent>)
ObjectPropertyAssertion(<iao:denotes> <ssvs:WGPS84
      LatitudeFieldComponent> <odt:realDatatype>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:WGPS84
      LatitudeFieldComponent> <ssvs:latitude>)
ClassAssertion(<odt:realFieldListSpecification> <ssvs:WGPS84
      ListFieldSpecification>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:WGPS84
      ListFieldSpecification> <ssvs:WGPS84LatitudeFieldComponent>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:WGPS84
      ListFieldSpecification> <ssvs:WGPS84LongitudeComponent>)
ClassAssertion(<odt:realFieldComponent> <ssvs:WGPS84
      LongitudeComponent>)
```

```
ObjectPropertyAssertion(<iao:denotes> <ssvs:WGPS84LongitudeComponent>
      <odt:realDatatype>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:WGPS84LongitudeComponent>
      <ssvs:longitude>)
ClassAssertion(<odt:recordOfReal> <ssvs:WGS84>)
AnnotationAssertion(<opis:hasIRI> <ssvs:behavioralAgent_classAxiom> <
     opis:BehavioralAgent>)
ClassAssertion(<opis:classAxiom> <ssvs:behavioralAgent_classAxiom>)
ClassAssertion(<odm:dataProcessingAlgorithm> <ssvs:
     calculateProbabilityPointOfInterestAlgorithm001>)
ObjectPropertyAssertion(<dul:hasPart> <ssvs:
     calculateProbabilityPointOfInterestAlgorithm001> <ssvs:
     calculateProbabilityPointOfInterestAlgorithmObjective001>)
ClassAssertion(<obi:dataTransformationObjective> <ssvs:
     calculateProbabilityPointOfInterestAlgorithmObjective001>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:
     calculateProbabilityPointOfInterestAlgorithmObjective001> <ssvs:
     HumanActivityOutputSpecification001>)
ClassAssertion(<odm:operator> <ssvs:
     calculateProbabilityPointOfInterestOperator001>)
ObjectPropertyAssertion(<obo:role_of> <ssvs:
     calculateProbabilityPointOfInterestOperator001> <ssvs:
     calculateProbabilityPointOfinterestAlgorithmImplementation001>)
ClassAssertion(<odm:algorithmImplementation> <ssvs:
     calculateProbabilityPointOfinterestAlgorithmImplementation001>)
ObjectPropertyAssertion(<obi:isConcretizationOf> <ssvs:
     calculateProbabilityPointOfinterestAlgorithmImplementation001> <
     ssvs:calculateProbabilityPointOfInterestAlgorithm001>)
ClassAssertion(<odm:descriptiveDataSpecification> <ssvs:
     descriptiveGeographiclocationDataSpecification_01>)
ObjectPropertyAssertion(<iao:isAbout> <ssvs:
     descriptiveGeographiclocationDataSpecification_01> <ssvs:WGS84>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:
     descriptiveGeographiclocationDataSpecification_01> <odm:
     descriptiveSpecification>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:
     descriptiveGeographiclocationDataSpecification_01> <ssvs:
     geographicLocationMappingSpecification_01>)
ClassAssertion(<opis:propertyMappingSpecification> <ssvs:
     geographicLocationMappingSpecification_01>)
ObjectPropertyAssertion(<iao:isAbout> <ssvs:
     geographicLocationMappingSpecification_01> <ssvs:
     geographicLocationProperty_dataPropertyAxiom>)
AnnotationAssertion(<opis:hasIRI> <ssvs:geographicLocationProperty_
     dataPropertyAxiom> <x-opis:geographicLocationProperty>)
ClassAssertion(<opis:dataPropertyAxiom> <ssvs:
     geographicLocationProperty_dataPropertyAxiom>)
ClassAssertion(<opis:semanticEntityMappingSpecification> <ssvs:
     humanActivityMappingSpecification001>)
ObjectPropertyAssertion(<iao:isAbout> <ssvs:
     humanActivityMappingSpecification001> <ssvs:behavioralAgent_
     classAxiom>)
ObjectPropertyAssertion(<iao:isAbout> <ssvs:
     humanActivityMappingSpecification001> <ssvs:humanActivity_
     classAxiom>)
ObjectPropertyAssertion(<iao:isAbout> <ssvs:
     humanActivityMappingSpecification001> <ssvs:isLocationOf_
     objectPropertyAxiom>)
ObjectPropertyAssertion(<iao:isAbout> <ssvs:
     humanActivityMappingSpecification001> <ssvs:pointOfInterest_
     classAxiom>)
ClassAssertion(<opis:semanticPerceptionDatatypeAttributeList> <ssvs:
     humanActivityPerceptionDatatypeAttributeList>)
ClassAssertion(<opis:AxiomsAttributeSpecification> <ssvs:
     humanActivityPerceptionDatatypeAxiomsAttribute001>)
ClassAssertion(<opis:SemanticPerceptionObjective> <ssvs:
     humanActivityPerceptionObjective_01>)
ClassAssertion(<opis:SemanticPerceptionProcess> <ssvs:
     humanActivityPerceptionProcess001>)
```

```
ObjectPropertyAssertion(<ro:has_part> <ssvs:
    humanActivityPerceptionProcess001> <ssvs:
    calculateProbabilityPointOfInterestAlgorithm001>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:
    humanActivityPerceptionProcess001> <ssvs:
    humanActivityPerceptionObjective_01>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:
    humanActivityPerceptionProcess001> <ssvs:
    selectPointOfInterestAlgorithm001>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:
    humanActivityPerceptionProcess001> <ssvs:stopDetectionAlgorithm
    001>)
AnnotationAssertion(<opis:hasIRI> <ssvs:humanActivity_classAxiom> <x-
    opis:PracticingSport>)
AnnotationAssertion(<opis:hasIRI> <ssvs:humanActivity_classAxiom> <
    opis:Behavior>)
ClassAssertion(<opis:classAxiom> <ssvs:humanActivity_classAxiom>)
AnnotationAssertion(<oboInOwl:hasURI> <ssvs:isLocationOf_
    objectPropertyAxiom> <dul:isLocationOf>)
ClassAssertion(<opis:objectPropertyAxiom> <ssvs:isLocationOf_
    objectPropertyAxiom>)
ClassAssertion(<odt:fieldIdentifier> <ssvs:latitude>)
ClassAssertion(<odt:fieldIdentifier> <ssvs:longitude>)
AnnotationAssertion(<opis:hasIRI> <ssvs:pointOfInterest_classAxiom> <
    x-opis:PointOfInterest>)
ClassAssertion(<opis:classAxiom> <ssvs:pointOfInterest_classAxiom>)
ClassAssertion(<odm:algorithmImplementation> <ssvs:
    selectPointOfInterestAlgoithmImplementation001>)
ObjectPropertyAssertion(<obi:isConcretizationOf> <ssvs:
    selectPointOfInterestAlgoithmImplementation001> <ssvs:
    selectPointOfInterestAlgorithm001>)
ObjectPropertyAssertion(<obi:hasQuality> <ssvs:
    selectPointOfInterestAlgoithmImplementation001> <ssvs:
    SpatialAccuracy001>)
ClassAssertion(<odm:dataProcessingAlgorithm> <ssvs:
    selectPointOfInterestAlgorithm001>)
ObjectPropertyAssertion(<dul:hasPart> <ssvs:
    selectPointOfInterestAlgorithm001> <ssvs:
    selectPointOfInterestAlgorithmObjective001>)
ObjectPropertyAssertion(<ro:precedes> <ssvs:
    selectPointOfInterestAlgorithm001> <ssvs:
    calculateProbabilityPointOfInterestAlgorithm001>)
ClassAssertion(<obi:dataTransformationObjective> <ssvs:
    selectPointOfInterestAlgorithmObjective001>)
ClassAssertion(<odm:operator> <ssvs:selectPointOfInterestOperator
    001>)
ObjectPropertyAssertion(<obo:role_of> <ssvs:
    selectPointOfInterestOperator001> <ssvs:
    selectPointOfInterestAlgoithmImplementation001>)
ObjectPropertyAssertion(<exact:hasInformation> <ssvs:
    selectPointOfInterestOperator001> <ssvs:
    SpatialAccuracyPoISelection001>)
ClassAssertion(<odm:dataProcessingAlgorithm> <ssvs:
    stopDetectionAlgorithm001>)
ObjectPropertyAssertion(<dul:hasPart> <ssvs:stopDetectionAlgorithm
    001> <ssvs:stopDetectionAlgorithmObjective001>)
ObjectPropertyAssertion(<ro:precedes> <ssvs:stopDetectionAlgorithm
    001> <ssvs:selectPointOfInterestAlgorithm001>)
ClassAssertion(<odm:algorithmImplementation> <ssvs:
    stopDetectionAlgorithmImplementation001>)
ObjectPropertyAssertion(<obi:isConcretizationOf> <ssvs:
    stopDetectionAlgorithmImplementation001> <ssvs:
    stopDetectionAlgorithm001>)
ObjectPropertyAssertion(<obi:hasQuality> <ssvs:
    stopDetectionAlgorithmImplementation001> <ssvs:SpatialAccuracy
    001>)
ClassAssertion(<obi:dataTransformationObjective> <ssvs:
    stopDetectionAlgorithmObjective001>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:
    stopDetectionAlgorithmObjective001> <ssvs:
    descriptiveGeographiclocationDataSpecification_01>)
```

```
ClassAssertion(<odm:operator> <ssvs:stopDetectionOperator001>)
ObjectPropertyAssertion(<obo:role_of> <ssvs:stopDetectionOperator001>
    <ssvs:stopDetectionAlgorithmImplementation001>)
ObjectPropertyAssertion(<exact:hasInformation> <ssvs:
    stopDetectionOperator001> <ssvs:SpatialAccuracyPoICalculation
    001>)
ClassAssertion(<opis:semanticEntityMappingSpecification> <ssvs:
    behavioralEntityMappingSpec001>)
ObjectPropertyAssertion(<iao:isAbout> <ssvs:
    behavioralEntityMappingSpec001> <ssvs:
    behavioralEntityForAccessControl001>)
ClassAssertion(<opis:semanticEntityMappingSpecification> <ssvs:
    behavioralEntityMappingSpec002>)
ObjectPropertyAssertion(<iao:isAbout> <ssvs:
    behavioralEntityMappingSpec002> <ssvs:geographicLocationProperty_
    dataPropertyAxiom>)
ClassAssertion(<odm:dataSpecification> <ssvs:dataSpec001>)
ObjectPropertyAssertion(<iao:isAbout> <ssvs:dataSpec001> <opis:
    semanticPerceptionDatatype001>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:dataSpec001> <ssvs:
    behavioralEntityForAccessControl001>)
ClassAssertion(<odm:dataSpecification> <ssvs:dataSpec002>)
ObjectPropertyAssertion(<iao:isAbout> <ssvs:dataSpec002> <opis:
    semanticPerceptionDatatype001>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:dataSpec002> <ssvs:
    behavioralEntityForAccessControl001>)
ClassAssertion(<odm:dataSpecification> <ssvs:dataSpec003>)
ObjectPropertyAssertion(<iao:isAbout> <ssvs:dataSpec003> <odt:
    realDatatype>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:dataSpec003> <ssvs:
    geographicLocationMappingSpecification_01>)
ClassAssertion(<odm:dataSpecification> <ssvs:dataSpec004>)
ObjectPropertyAssertion(<iao:isAbout> <ssvs:dataSpec004> <odt:
    realDatatype>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:dataSpec004> <ssvs:
    geographicLocationMappingSpecification_01>)
ClassAssertion(<iao:objectiveSpecification> <ssvs:objSpec001>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:objSpec001> <ssvs:
    dataSpec001>)
ClassAssertion(<iao:objectiveSpecification> <ssvs:objSpec002>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:objSpec002> <ssvs:
    dataSpec002>)
ClassAssertion(<iao:objectiveSpecification> <ssvs:objSpec003>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:objSpec003> <ssvs:
    dataSpec003>)
ClassAssertion(<iao:objectiveSpecification> <ssvs:objSpec004>)
ObjectPropertyAssertion(<ro:has_part> <ssvs:objSpec004> <ssvs:
    dataSpec004>)
```

10.3 RETRIEVEPPVFORVS

```
PREFIX opis: <https://../thiagomoreirac/opis/master/opis.owl#>
PREFIX ppol: <https://..iagomoreirac/opis/master/privacy_policy.owl#>
PREFIX obi: <http://purl.obolibrary.org/obo/>
PREFIX ro: <http://www.obofoundry.org/ro/ro.owl#>
PREFIX iao: <http://purl.obolibrary.org/obo/>
PREFIX odm: <http://kt.ijs.si/panovp/OntoDM#>
PREFIX odt: <http://kt.ijs.si/panovp/OntoDT#>
PREFIX ssn: <http://purl.oclc.org/NET/ssnx/ssn#>
SELECT ?iPPVS ?iOntologyAxiom ?cProperty ?cFeatureOfInterest ?
    iDatatype WHERE {{
#=== Privacy Policy Condition =======
?privacyPolicyCondition ppol:includesPrivacyPreservingVirtualSensor ?
    iPPVS.
?privacyPolicyCondition dul:isSettingFor ?iDirectClassification.
?privacyPolicyCondition a ?PPC. ?PPC rdfs:subClassOf+ ppol:
    PrivacyPolicyCondition.
```

```
#=== Transversal Classification ====
?iDirectClassification a/rdfs:subClassOf* ?cTransversalClassification
    .
?cTransversalClassification rdfs:subClassOf* dul:Concept; ^a ?
    iTransversalClassification.
#=== Concrete Classification =======
?iTransversalClassification dul:classifies ?cTCOntologyAxiom.
?cTCOntologyAxiom a/rdfs:subClassOf+ opis:OntologyAxiom; obo:hasURI ?
    cTransversalClassifiedAxiom.
?cTransversalClassifiedAxiom (rdfs:subClassOf|owl:equivalentClass|^
    owl:equivalentClass)+ opis:BehavioralEntity.
?cTransversalClassifiedAxiom ((rdfs:subClassOf/(owl:someValuesFrom|
    owl:allValuesFrom|owl:onClass))|((owl:unionOf/rdf:rest*/rdf:first
    )*))* ?cClassifiedAxiom.
?iClassifiedAxiom a/(rdfs:subClassOf|^owl:equivalentClass|owl:
    equivalentClass)* ?cClassifiedAxiom.
#=== Output VS ===============
?iVirtualSensor a/rdfs:subClassOf* opis:VirtualSensor.
?iVirtualSensor obi:isConcretizationOf ?iSPP. ?iSPP a/rdfs:subClassOf
    * opis:SemanticPerceptionProcess.
?iSPP ro:has_part|dul:hasPart ?iAlgorithm. ?iAlgorithm a/rdfs:
    subClassOf iao:iao:algorithm.
?iAlgorithm ro:has_part|dul:hasPart ?iObjectiveSpec. ?iObjectiveSpec
    a/rdfs:subClassOf* iao:objectSpecification.
?iObjectiveSpec ro:has_part|dul:hasPart ?iOutputDataSpec. ?
    iOutputDataSpec a/rdfs:subClassOf* odm:OntoDM_000168.
?iOutputDataSpec ro:has_part|dul:hasPart ?iMapSpec. ?iMapSpec a/rdfs:
    subClassOf* odm:mappingSpecification.
?iOutputDataSpec iao:iao:isAbout ?iDatatype. ?iDatatype a/rdfs:
    subClassOf* odt:datatype.
?iMapSpec iao:iao:isAbout ?iOntologyAxiom. ?iOntologyAxiom a/rdfs:
    subClassOf* opis:OntologyAxiom.
?iOntologyAxiom opis:hasIRI ?cOntologyAxiom. ?cOntologyAxiom rdfs:
    subClassOf* ?IRIType.
#=== Input VS  ===============
?iVirtualSensor a/rdfs:subClassOf* opis:VirtualSensor.
?iVirtualSensor obi:isConcretizationOf ?iSPP. ?iSPP a/rdfs:subClassOf
    * opis:SemanticPerceptionProcess.
?iSPP ro:has_part|dul:hasPart ?iAlgorithm. ?iAlgorithm a/rdfs:
    subClassOf iao:iao:algorithm.
?iAlgorithm ro:has_part|dul:hasPart ?iObjectiveSpec. ?iObjectiveSpec
    a/rdfs:subClassOf* iao:objectSpecification.
?iObjectiveSpec ro:has_part|dul:hasPart ?iInputSpec. ?iInputSpec a/
    rdfs:subClassOf* odm:descriptiveDataSpecification.
?iInputSpec ro:has_part|dul:hasPart ?iMapSpec. ?iMapSpec a/rdfs:
    subClassOf* odm:mappingSpecification.
?iInputSpec iao:iao:isAbout ?iDatatype. ?iDatatype a/rdfs:subClassOf*
     odt:datatype.
?iMapSpec iao:iao:isAbout|(iao:iao:isAbout/opis:hasIRI) ?cProperty. ?
    cProperty rdfs:subClassOf* ssn:Property.
OPTIONAL {?cProperty ssn:isPropertyOf ?cFeatureOfInterest. ?
    cFeatureOfInterest rdfs:subClassOf* ssn:FeatureOfInterest.}
#=== Input PPPVS ===============
?iPPVS a/rdfs:subClassOf* opis:VirtualSensor.
?iPPVS obi:isConcretizationOf ?iSPP2. ?iSPP2 a/rdfs:subClassOf* opis:
    SemanticPerceptionProcess.
?iSPP2 ro:has_part|dul:hasPart ?iAlgorithm2. ?iAlgorithm2 a/rdfs:
    subClassOf iao:iao:algorithm.
?iAlgorithm2 ro:has_part|dul:hasPart ?iObjectiveSpec2. ?
    iObjectiveSpec2 a/rdfs:subClassOf* iao:objectiveSpecification.
?iObjectiveSpec2 ro:has_part|dul:hasPart ?iInputSpec2. ?iInputSpec2 a
    /rdfs:subClassOf* odm:descriptiveDataSpecification.
?iInputSpec2 ro:has_part|dul:hasPart ?iMapSpec2. ?iMapSpec2 a/rdfs:
    subClassOf* odm:mappingSpecification.
?iInputSpec2 iao:iao:isAbout ?iDatatype. ?iDatatype a/rdfs:subClassOf
    * odt:datatype.
?iMapSpec2 iao:iao:isAbout|(iao:iao:isAbout/opis:hasIRI) ?cProperty.
    ?cProperty rdfs:subClassOf* ssn:Property.
OPTIONAL {?cProperty ssn:isPropertyOf ?cFeatureOfInterest. ?
    cFeatureOfInterest rdfs:subClassOf* ssn:FeatureOfInterest.}
#=== RetrievePPVSForVS ===============
```

```
?cOntologyAxiom rdfs:subClassOf*/^a ?iClassifiedAxiom.
}
FILTER (?IRIType IN (ssn:FeatureOfInterest, ssn:Property))
FILTER (?iVirtualSensor != ?iPPVS)
FILTER (?iVirtualSensor = :ID)
} LIMIT :LIMIT
```

BIBLIOGRAPHY

[1] Shancang Li, Li Da Xu, and Shanshan Zhao. « The internet of things: a survey. » In: *Information Systems Frontiers* 17.2 (2015), pp. 243–259. ISSN: 1387-3326. DOI: 10.1007/s10796-014-9492-7. arXiv: arXiv:1011.1669v3. URL: http://link.springer.com/10.1007/s10796-014-9492-7.

[2] Prith Banerjee, Richard Friedrich, Cullen Bash, Patrick Goldsack, Bernardo A. Huberman, John Manley, Chandrakant Patel, Parthasarathy Ranganathan, and Alistair Veitch. « Everything as a service: Powering the new information economy. » In: *Computer* 44.3 (2011), pp. 36–43. ISSN: 00189162. DOI: 10.1109/MC.2011.67.

[3] David D'iaz Pardo de Vera, Álvaro Siüenza Izquierdo, Jesús Bernard Vercher, and Luis Alfonso Hernández Gómez. « A Ubiquitous Sensor Network Platform for Integrating Smart Devices into the Semantic Sensor Web. » In: *Sensors* 14.6 (2014), pp. 10725–10752. ISSN: 1424-8220. DOI: 10.3390/s140610725. URL: http://www.mdpi.com/1424-8220/14/6/10725/.

[4] S. Sicari, A. Rizzardi, L.A. Grieco, and A. Coen-Porisini. « Security, privacy and trust in Internet of Things: The road ahead. » In: *Computer Networks* 76 (2015), pp. 146–164. ISSN: 13891286. DOI: 10.1016/j.comnet.2014.11.008. URL: http://dx.doi.org/10.1016/j.comnet.2014.11.008http://linkinghub.elsevier.com/retrieve/pii/S1389128614003971.

[5] European Commission. *Proposal for a Regulation of the European parliament and of the council on the protection of individuals with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation)*. 2012. DOI: 2012/0011(COD). arXiv: arXiv:1011.1669v3. URL: http://ec.europa.eu/justice/data-protection/document/review2012/com{_}2012{_}11{_}en.pdf.

[6] European Commision. *EU Commission and United States agree on new framework for transatlantic data flows: EU-US Privacy Shield*. 2016. URL: http://europa.eu/rapid/press-release{_}IP-16-216{_}en.htm (visited on 09/01/2016).

[7] Elena Simperl, Kieron O'Hara, and Gomer Richard. *Open Data and Privacy, An analytical report*. Tech. rep. Eletronics and Computer Sience, University of Southampton, 2016. URL: http://www.europeandataportal.eu/.

[8] FTC. « IoT Privacy & Security in a Connected World. » In: January (2015), p. 71.

[9] Isabelle Oomen and Ronald Leenes. « Privacy Risk Perceptions and Privacy Protection Strategies. » In: *Policies and Research in Identity Management* 261 (2008), pp. 121–138. ISSN: 15715736. DOI: 10.1007/978-0-387-77996-6.

[10] Alessandro Acquisti, Laura Brandimarte, and George Loewenstein. « Privacy and human behavior in the age of information. » In: *Science* 347.6221 (2015), pp. 509–514.

[11] Daniel J. Solove. « A Taxonomy of Privacy. » In: *University of Pennsylvania Law Review* 154.3 (2006), pp. 477–560. ISSN: 00419907. DOI: 10.2307/40041279. URL: http://www.jstor.org/stable/40041279{\%}5Cnhttps://www.law.upenn.edu/journals/lawreview/articles/volume154/issue3/Solove154U.Pa.L.Rev.477(2006).pdf.

[12] Samuel D Warren and Louis D Brandeis. « The Right to Privacy. » In: *Harvard Law Review* 4.5 (1890), pp. 193–220. ISSN: 0017811X. DOI: 10.2307/1321160. URL: http://www.jstor.org/stable/1321160.

[13] Alessandro Mantelero. « The EU Proposal for a General Data Protection Regulation and the roots of the right to be forgotten. » In: *Computer Law and Security Review* 29.3 (2013), pp. 229–235. ISSN: 02673649. DOI: 10.1016/j.clsr.2013.03.010.

[14] Stephen B. Wicker and Dawn E. Schrader. « Privacy-aware design principles for information networks. » In: *Proceedings of the IEEE* 99.2 (2011), pp. 330–350. ISSN: 00189219. DOI: 10.1109/JPROC.2010.2073670.

[15] Helen Nissenbaum. « Privacy as contextual integrity. » In: *Washington Law Review* 79.1 (2004), p. 66. ISSN: 09700420. DOI: 10.1109/SP.2006.32. arXiv: arXiv:1011.1669v3.

[16] Julie E. Cohen. « Examined Lives: Informational Privacy and the Subject as Object. » In: *Stanford Law Review* 52 (2000), pp. 1373–1438. ISSN: 00389765. DOI: 10.2307/1229517.

[17] Michel Foucault. *Discipline and Punish: The Birth of the Prison.* 1978. DOI: 10.2307/2065008. arXiv: 9809069v1 [arXiv:gr-qc].

[18] A Poulter. « Encyclopedia of Library and Information Science. » In: *Journal of the Medical Library Association* 92 (2003), p. 106. URL: http://strathprints.strath.ac.uk/2540/.

[19] Russell L Ackoff. « From data to wisdom. » In: *Journal of Applied Systems Analysis* 16.1 (1989), pp. 3–9. ISSN: 12345792. DOI: citeulike-article-id:6930744.

[20] Ofer Bergman, Richard Boardman, Jacek Gwizdka, and William Jones. « Personal information management. » In: *Extended abstracts of the 2004 conference on Human factors and computing systems CHI 04* (2004), p. 1598. DOI: 10.1145/985921.986164. URL: http://portal.acm.org/citation.cfm?doid=985921.986164.

[21] Longbing Cao. « In-depth behavior understanding and use: The behavior informatics approach. » In: *Information Sciences* 180.17 (2010), pp. 3067–3085. ISSN: 00200255. DOI: 10.1016/j.ins.2010.03.025. URL: http://linkinghub.elsevier.com/retrieve/pii/S0020025510001374.

[22] Seda Gürses and Jose M. Del Alamo. « Privacy Engineering: Shaping an Emerging Field of Research and Practice. » In: *IEEE Security and Privacy* 14.2 (2016), pp. 40–46. ISSN: 15584046. DOI: 10.1109/MSP.2016.37.

[23] Xiang Sheng, Xuejie Xiao, Jian Tang, and Guoliang Xue. « Sensing as a service: A cloud computing system for mobile phone sensing. » In: *Proceedings of IEEE Sensors* (2012). ISSN: 1930-0395. DOI: 10.1109/ICSENS.2012.6411516.

[24] Sarah Spiekermann and Lorrie Faith Cranor. « Engineering privacy. » In: *IEEE Transactions on Software Engineering* 35.1 (2009), pp. 67–82. ISSN: 00985589. DOI: 10.1109/TSE.2008.88.

[25] Paul Ohm. « Broken Promises of Privacy: Responding to the Surprising Failure of Anoymization. » In: *UCLA Law Review* 57 (2010), pp. 1701–1777. ISSN: 00415650. DOI: 10.2139/ssrn.1450006.

[26] S Abdelwahab, B Hamdaoui, M Guizani, and T Znati. « Cloud of Things for Sensing as a Service: Sensing Resource Discovery and Virtualization. » In: *2015 IEEE Global Communications Conference (GLOBECOM)* (2015), pp. 1–7. DOI: 10.1109/GLOCOM.2015.7417252.

[27] Jean-paul Calbimonte, Sofiane Sarni, Julien Eberle, and Karl Aberer. « XGSN: an open-source semantic sensing middleware for the web of things. » In: *Joint Proceedings of the 6th International Workshop on the Foundations, Technologies and Applications of the Geospatial Web, TC*. 2014, pp. 51–66. URL: http://knoesis.org/ssn2014/paper{_}8.pdf.

[28] K. Ashton. *That 'internet of things' thing in the real world, things matter more than ideas*. 2009. URL: http://www.rfidjournal.com/article/print/4986 (visited on 03/30/2016).

[29] International Telecommunication Union. « Overview of the Internet of things. » In: *Series Y: Global information infrastructure, internet protocol aspects and next-generation networks - Frameworks and functional architecture models* (2012), p. 22.

[30] Gerd Kortuem, Fahim Kawsar, Daniel Fitton, and Vasughi Sundramoorthy. « Smart Objects as Building Blocks for the Internet of Things. » In: *IEEE Computer Society* 10 (2010), pp. 1089–7801. ISSN: 1089-7801. DOI: 10.1109/MIC.2009.143.

[31] Andrew Whitmore, Anurag Agarwal, and Li Da Xu. « The Internet of Things—A survey of topics and trends. » In: *Information Systems Frontiers* 17.2 (2015), pp. 261–274. ISSN: 1387-3326. DOI: 10.1007/s10796-014-9489-2. URL: http://link.springer.com/10.1007/s10796-014-9489-2.

[32] Panagiotis Demestichas, Andreas Georgakopoulos, Dimitrios Karvounas, Kostas Tsagkaris, Vera Stavroulaki, Jianmin Lu, Chunshan Xiong, and Jing Yao. « 5G on the Horizon: Key Challenges for the Radio-Access Network. » In: *IEEE Vehicular Technology Magazine* 8.3 (2013), pp. 47–53. ISSN: 1556-6072. DOI: 10.1109/MVT.2013.2269187. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6568922.

[33] Noboru Koshizuka and Ken Sakamura. « Ubiquitous ID: Standards for Ubiquitous Computing and the Internet of Things. » In: *IEEE Pervasive Computing* 9.4 (2010), pp. 98–101. ISSN: 1536-1268. DOI: 10.1109/MPRV.2010.87. URL: http://ieeexplore.ieee.org/xpl/freeabs{_}all.jsp?isnumber=5586685{\&}arnumber=5586696http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5586696.

[34] Alessio Botta, Walter De Donato, Valerio Persico, and Antonio Pescapé. « Integration of Cloud computing and Internet of Things: A survey. » In: *Future Generation Computer Systems* 56 (2016), pp. 684–700. ISSN: 0167739X. DOI: 10.1016/j.future.2015.09.021. URL: http://dx.doi.org/10.1016/j.future.2015.09.021.

[35] Maria Rita Palattella, Nicola Accettura, Xavier Vilajosana, Thomas Watteyne, Luigi Alfredo Grieco, Gennaro Boggia, and Mischa Dohler. « Standardized protocol stack for the internet of (important) things. » In: *IEEE Communications Surveys and Tutorials* 15.3 (2013), pp. 1389–1406. ISSN: 1553877X. DOI: 10.1109/SURV.2012.111412.00158.

[36] Li Da Xu, Wu He, and Shancang Li. « Internet of things in industries: A survey. » In: *IEEE Transactions on Industrial Informatics* 10.4 (2014), pp. 2233–2243. ISSN: 15513203. DOI: 10.1109/TII.2014.2300753.

[37] Gil Reiter. « Wireless connectivity for the Internet of Things. » In: *Texas Instrument White Paper* (2014), pp. 1–13.

[38] Ala Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. « Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. » In: *IEEE Communications Surveys and Tutorials* 17.4 (2015), pp. 2347–2376. ISSN: 1553877X. DOI: 10.1109/COMST.2015.2444095.

[39] Matthew Gigli. « Internet of Things: Services and Applications Categorization. » In: *Advances in Internet of Things* 01.02 (2011), pp. 27–31. ISSN: 2161-6817. DOI: 10.4236/ait.2011.12004.

[40] Michael Compton et al. « The SSN ontology of the W3C semantic sensor network incubator group. » In: *Web Semantics: Science, Services and Agents on the World Wide Web* 17 (2012), pp. 25–32. ISSN: 15708268. DOI: 10.1016/j.websem.2012.05.003. URL: http://dx.doi.org/10.1016/j.websem.2012.05.003http://linkinghub.elsevier.com/retrieve/pii/S1570826812000571.

[41] Jayavardhana Gubbi, Rajkumar Buyya, Slaven Marusic, and Marimuthu Palaniswami. « Internet of Things (IoT): A Vision , Architectural Elements , and Future Directions. » In: *Future Generation Computer Systems* 29.7 (2013), pp. 1645–1660. ISSN: 0167739X. DOI: 10.1016/j.future.2013.01.010. arXiv: 1207.0203.

[42] Charu C Aggarwal, Naveen Ashish, and Amit Sheth. « The Internet of Things: A Survey from the Data-Centric Perspective. » In: *Managing and Mining Sensor Data*. Boston, MA: Springer US, 2013, pp. 383–428. ISBN: 978-1461463085. DOI: 10 . 1007 / 978 - 1 - 4614 - 6309 - 2 _ 12. URL: http://link.springer.com/10.1007/978-1-4614-6309-2{_}12.

[43] Ghofrane Fersi. « Middleware for internet of things: A study. » In: *Proceedings - IEEE International Conference on Distributed Computing in Sensor Systems, DCOSS 2015* (2015), pp. 230–235. ISSN: 09763252. DOI: 10.1109/DCOSS.2015.43.

[44] Manuel Díaz, Cristian Martín, and Bartolomé Rubio. « State-of-the-art, challenges, and open issues in the integration of Internet of things and cloud computing. » In: *Journal of Network and Computer Applications* 67 (2015), pp. 99–117. ISSN: 10958592. DOI: 10.1016/j.jnca.2016.01.010. URL: http://dx.doi.org/10.1016/j.jnca.2016.01.010.

[45] Peter Mell and Timothy Grance. « The NIST Definition of Cloud Computing Recommendations of the National Institute of Standards and Technology. » In: *National Institute of Standards and Technology, Information Technology Laboratory* 145 (2011), p. 7. ISSN: 1472-0213. DOI: 10.1136/emj.2010.096966. arXiv: 2305-0543. URL: http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf.

[46] L Arockiam, S Monikandan, G Parthasarathy, and A Properties Cloud. « Cloud Computing : A Survey. » In: *International Journal of Internet Computing* 1.2 (2011), pp. 26–33. URL: http://interscience.in/IJIC{_}Vol1Iss2/paper5.pdf.

[47] Qi Zhang, Lu Cheng, and Raouf Boutaba. « Cloud computing: State-of-the-art and research challenges. » In: *Journal of Internet Services and Applications* 1.1 (2010), pp. 7–18. ISSN: 18674828. DOI: 10.1007/s13174-010-0007-6. arXiv: S0167739X10002554.

[48] Yucong Duan, Guohua Fu, Nianjun Zhou, Xiaobing Sun, Nanjangud C. Narendra, and Bo Hu. « Everything as a Service (XaaS) on the Cloud: Origins, Current and Future Trends. » In: *Proceedings - 2015 IEEE 8th International Conference on Cloud Computing, CLOUD 2015* (2015), pp. 621–628. DOI: 10.1109/CLOUD.2015.88.

[49] Zhibo Pang, Qiang Chen, Weili Han, and Lirong Zheng. « Value-centric design of the internet-of-things solution for food supply chain: Value creation, sensor portfolio and information fusion. » In: *Information Systems Frontiers* 17.2

(2015), pp. 289–319. ISSN: 13873326. DOI: 10.1007/s10796-012-9374-9.

[50] OECD. « Thirty Years After the Oecd Privacy Guidelines. » In: (2011), pp. 1–111. DOI: 10.1787/5kgf09z90c31-en.

[51] 32nd International Conference of Data Protection and Privacy Commissioners. *Resolution on Privacy by Design*. Jerusalem, Israel, 2010. URL: https://secure.edps.europa.eu/EDPSWEB/webdav/site/mySite/shared/Documents/Cooperation/Conference{_}int/10-10-27{_}Jerusalem{_}Resolutionon{_}PrivacybyDesign{_}EN.pdf.

[52] Marc Langheinrich. « Privacy by Design - Principles of Privacy-Aware Ubiquitous Systems. » In: *3rd international conference on Ubiquitous Computing* (2001), pp. 273–291. URL: http://dl.acm.org/citation.cfm?id=647987.741336.

[53] Thibaud Antignac, D Le Métayer, and Daniel Le M. « Privacy by Design: From Technologies to Architectures. » In: *Privacy Technologies and Policy* (2014), pp. 1–17. DOI: 10.1007/978-3-319-06749-0_1. arXiv: arXiv:1410.0030v1. URL: http://www.ipc.on.ca/images/Resources/2009-06-23-TrustEconomics.pdf{\%}5Cnhttp://link.springer.com/chapter/10.1007/978-3-319-06749-0{_}1.

[54] Anco Hundepool, Josep Domingo-ferrer, Luisa Franconi, Sarah Giessing, Rainer Lenz, Jane Longhurst, Eric Schulte Nordholt, Giovanni Seri, and Peter-paul De Wolf. *Handbook on Statistical Disclosure Control*. March. CENDEX, 2007, pp. 1–208.

[55] Salvatore Distefano, Giovanni Merlino, and Antonio Puliafito. « A utility paradigm for IoT: The sensing Cloud. » In: *Pervasive and Mobile Computing* 20 (2015), pp. 127–144. ISSN: 15741192. DOI: 10.1016/j.pmcj.2014.09.006.

[56] Yongrui Qin, Quan Z. Sheng, Nickolas J G Falkner, Schahram Dustdar, Hua Wang, and Athanasios V. Vasilakos. « When things matter: A survey on data-centric internet of things. » In: *Journal of Network and Computer Applications* 64 (2016), pp. 137–153. ISSN: 10958592. DOI: 10.1016/j.jnca.2015.12.016. arXiv: 1407.2704. URL: http://dx.doi.org/10.1016/j.jnca.2015.12.016.

[57] Marica Amadeo, Claudia Campolo, Jose Quevedo, Daniel Corujo, Antonella Molinaro, Antonio Iera, Rui L. Aguiar, and Athanasios V. Vasilakos. « Information-centric networking for the internet of things: Challenges and opportunities. » In: *IEEE Network* 30.2 (2016), pp. 92–100. ISSN: 08908044. DOI: 10.1109/MNET.2016.7437030. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7437030.

[58] Amit Sheth, Pramod Anantharam, and Cory Henson. « Physical-cyber-social computing: An early 21st century approach. » In: *IEEE Intelligent Systems* 28.1 (2013), pp. 78–82. ISSN: 15411672. DOI: 10.1109/MIS.2013.20.

[59] Bin Guo, Daqing Zhang, Zhu Wang, Zhiwen Yu, and Xingshe Zhou. « Opportunistic IoT: Exploring the harmonious interaction between human and the internet of things. » In: *Journal of Network and Computer Applications* 36.6 (2013), pp. 1531–1539. ISSN: 10848045. DOI: 10.1016/j.jnca.2012.12.028. URL: http://dx.doi.org/10.1016/j.jnca.2012.12.028.

[60] Payam Barnaghi, Wei Wang, Cory Henson, and Kerry Taylor. « Semantics for the Internet of Things. » In: *International Journal on Semantic Web and Information Systems* 8.1 (2012), pp. 1–21. ISSN: 1552-6283. DOI: 10.4018/jswis.2012010101. URL: http://services.igi-global.com/resolvedoi/resolve.aspx?doi=10.4018/jswis.2012010101.

[61] Geoffrey C. Fox, Supun Kamburugamuve, and Ryan D. Hartman. « Architecture and measured characteristics of a cloud based internet of things. » In: *Proceedings of the 2012 International Conference on Collaboration Technologies and Systems, CTS 2012* (2012), pp. 6–12. DOI: 10.1109/CTS.2012.6261020.

[62] Byung-Gon Chun and Petros Maniatis. « Dynamically partitioning applications between weak devices and clouds. » In: *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing & Services Social Networks and Beyond - MCS '10*. New York, New York, USA: ACM Press, 2010, pp. 1–5. ISBN: 9781450301558. DOI: 10.1145/1810931.1810938. URL: http://portal.acm.org/citation.cfm?doid=1810931.1810938.

[63] Andreas Fischer, Juan Felipe Botero, Michael Till Beck, Hermann De Meer, and Xavier Hesselbach. « Virtual network embedding: A survey. » In: *IEEE Communications Surveys and Tutorials* 15.4 (2013), pp. 1888–1906. ISSN: 1553877X. DOI: 10.1109/SURV.2013.013013.00155.

[64] Sanjay Madria, Vimal Kumar, and Rashmi Dalvi. « Sensor Cloud : A Cloud of Virtual Sensors. » In: (2014). DOI: 10.1109/NBiS.2010.32.4..

[65] M. Botts, M. Botts, G. Percivall, G. Percivall, C. Reed, C. Reed, J. Davidson, and J. Davidson. « OGC (R) Sensor Web Enablement: Overview and High Level Architecture. » In: *Lecture Notes In Computer Science* 4540.December (2013), pp. 175–190. DOI: 10.1007/978-3-540-79996-2. URL: http://www.springerlink.com/index/ux1224j76264g8j4.pdf.

[66] Flavio Bonomi, Rodolfo Milito, Preethi Natarajan, and Jiang Zhu. « Big Data and Internet of Things: A Roadmap for Smart Environments. » In: 546.August 2015 (2014), pp. 169–186. ISSN: 1860949X. DOI: 10.1007/978-3-319-05029-4. URL: http://link.springer.com/10.1007/978-3-319-05029-4.

[67] K. Mohammed Hoda and Amany F. Soliman. « Data Stream Mining. » In: *Methodology* 8.May (2011), pp. 127–141. ISSN: 21508097. DOI: 10.1007/978-0-387-09823-4_39. URL: http://dspace.cusat.ac.in/jspui/handle/123456789/3616.

[68] Cory Henson, Amit Sheth, and Krishnaprasad Thirunarayan. « Semantic Perception: Converting Sensory Observations to Abstractions. » In: *IEEE Internet Computing* 16.2 (2012), pp. 26–34. ISSN: 1089-7801. DOI: 10.1109/MIC.2012.20. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6133260.

[69] Qihui Wu, Guoru Ding, Yuhua Xu, Shuo Feng, Zhiyong Du, Jinlong Wang, and Keping Long. « Cognitive Internet of Things: A New Paradigm Beyond Connection. » In: *IEEE Internet of Things Journal* 1.2 (2014), pp. 129–143. ISSN: 2327-4662. DOI: 10.1109/JIOT.2014.2311513. arXiv: arXiv:1403.2498v1. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6766209.

[70] Fernando Roda and Estanislao Musulin. « An ontology-based framework to support intelligent data analysis of sensor measurements. » In: *Expert Systems with Applications* 41.17 (2014), pp. 7914–7926. ISSN: 09574174. DOI: 10.1016/j.eswa.2014.06.033. URL: http://linkinghub.elsevier.com/retrieve/pii/S0957417414003741.

[71] Swarup Chandra, Justin Sahs, Latifur Khan, Bhavani Thuraisingham, and Charu Aggarwal. « Stream Mining Using Statistical Relational Learning. » In: *2014 IEEE International Conference on Data Mining* (2014), pp. 743–748. ISSN: 1550-4786. DOI: 10.1109/ICDM.2014.144. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7023394.

[72] Hao Yue, Linke Guo, Ruidong Li, Hitoshi Asaeda, and Yuguang Fang. « DataClouds: Enabling community-based data-centric services over the internet of things. » In: *IEEE Internet of Things Journal* 1.5 (2014), pp. 472–482. ISSN: 23274662. DOI: 10.1109/JIOT.2014.2353629.

[73] Charith Perera, Arkady Zaslavsky, Peter Christen, and Dimitrios Georgakopoulos. « Sensing as a Service Model for Smart Cities Supported by Internet of Things. » In: *Transactions on emerging telecommunications technologies* (2014). ISSN: 15418251. DOI: 10.1002/ett. arXiv: arXiv:1307.8198v1.

[74] M. Srivastava, T. Abdelzaher, and B. Szymanski. « Human-centric sensing. » In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 370.1958 (2012), pp. 176–197. ISSN: 1364-503X. DOI: 10.1098/rsta.2011.0244. URL: http://rsta.royalsocietypublishing.org/cgi/doi/10.1098/rsta.2011.0244.

[75] Nicholas D. Lane, Emiliano Miluzzo, Hong Lu, Daniel Peebles, Tanzeem Choudhury, and Andrew T. Campbell. « A survey of mobile phone sensing. » In: *IEEE Communications Magazine* 48.9 (2010), pp. 140–150. ISSN: 01636804. DOI: 10.1109/MCOM.2010.5560598. arXiv: 10 [0163-6804].

[76] Burak Kantarci and Hussein T. Mouftah. « Sensing services in cloud-centric Internet of Things: A survey, taxonomy and challenges. » In: *2015 IEEE International Conference on Communication Workshop, ICCW 2015* (2015), pp. 1865–1870. DOI: 10.1109/ICCW.2015.7247452.

[77] Eleonora Borgia. « The internet of things vision: Key features, applications and open issues. » In: *Computer Communications* 54 (2014), pp. 1–31. ISSN: 01403664. DOI: 10.1016/j.comcom.2014.09.008. arXiv: 1207.0203. URL: http://dx.doi.org/10.1016/j.comcom.2014.09.008.

[78] Hyojoon Kim and Nick Feamster. « Improving network management with software defined networking. » In: *IEEE Communications Magazine* 51.2 (2013), pp. 114–119. ISSN: 01636804. DOI: 10.1109/MCOM.2013.6461195.

[79] Oracle White Paper, Digital Twins, and F O R Iot. *Digital Twins for IoT Applications A Comprehensive Approach to Implementing IoT Digital Twins*. Tech. rep. January. 2017.

[80] Dimitri Volkmann. *The Rise of Digital Twins*. 2016.

[81] Wei Wang, Payam Barnaghi, Gilbert Cassar, Frieder Ganz, and Pirabakaran Navaratnam. « Semantic sensor service networks. » In: *Proceedings of IEEE Sensors* (2012). ISSN: 1930-0395. DOI: 10.1109/ICSENS.2012.6411490.

[82] Cory A. Henson, Josh K. Pschorr, Amit P. Sheth, and Krishnaprasad Thirunarayan. « SemSOS: Semantic sensor observation service. » In: *2009 International Symposium on Collaborative Technologies and Systems, CTS 2009* (2009), pp. 44–53. DOI: 10.1109/CTS.2009.5067461.

[83] Michael Compton, David Corsar, and Kerry Taylor. « Sensor data provenance: SSNO and PROV-O together at last. » In: *Terra Cognita and Semantic Sensor, Networks* (2014), pp. 67–82. URL: http://ceur-ws.org/Vol-1401/paper-05.pdf.

[84] Alessandro Margara, Jacopo Urbani, Frank Van Harmelen, and Henri Bal. « Streaming the Web: Reasoning over dynamic data. » In: *Journal of Web Semantics* 25 (2014), pp. 24–44. ISSN: 15708268. DOI: 10.1016/j.websem.2014.02.001.

[85] Davide Barbieri, Daniele Braga, Stefano Ceri, Emanuele Della Valle, Yi Huang, Volker Tresp, Achim Rettinger, and Hendrik Wermser. « Deductive and inductive stream reasoning for semantic social media analytics. » In: *IEEE Intelligent Systems* 25.6 (2010), pp. 32–41. ISSN: 15411672. DOI: 10.1109/MIS.2010.142.

[86] Darko Anicic and Paul Fodor. « EP-SPARQL: a unified language for event processing and stream reasoning. » In: *Proceedings of the 20th . . .* (2011), pp. 635–644. ISSN: 0018-8670. DOI: 10.1147/sj.433.0598. URL: http://dl.acm.org/citation.cfm?id=1963495.

[87] Danh Le Phuoc, Minh Dao-Tran, Josiane Xavier Parreira, and Manfred Hauswirth. « A Native and Adaptive Approach for Unified Processing of Linked Streams and Linked Data. » In: *The Semantic Web - {ISWC} 2011 - 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part {I}* 7031.24761 (2011), pp. 370–388. DOI: 10.1007/978-3-642-25073-6_24. URL: http://dx.doi.org/10.1007/978-3-642-25073-6{_}24.

[88] G Cugola and A Margara. « Processing flows of information: From data stream to complex event processing. » In: *ACM Computing Surveys (CSUR)* 44.3 (2012), p. 15.

[89] Amit Sheth. « Internet of Things to Smart IoT Through Semantic, Cognitive, and Perceptual Computing. » In: *IEEE Intelligent Systems* 31.2 (2016), pp. 108–112. ISSN: 15411672. DOI: 10.1109/MIS.2016.34.

[90] Cory Andrew Henson. « A Semantics-based Approach to Machine Perception. » Doctoral Dissertation. Wright State University, 2013, p. 166. URL: http://rave.ohiolink.edu/etdc/view?acc{_}num=wright1387645909.

[91] Harshal Patni, Satya S. Sahoo, Cory Henson, and Amit Shetli. « Provenance aware linked sensor data. » In: *CEUR Workshop Proceedings* 576 (2010). ISSN: 16130073.

[92] Danh Le-phuoc, Josiane Xavier Parreira, and Manfred Hauswirth. « Linked Stream Data Processing. » In: *Reasoning Web. Semantic Technologies for Advanced Query Answering 8th International Summer School* (2012), pp. 245–289. ISSN: 03029743. DOI: 10.1007/978-3-642-33158-9_7.

[93] Karl Aberer, Manfred Hauswirth, and Ali Salehi. « Infrastructure for data processing in large-scale interconnected sensor networks. » In: *Proceedings - IEEE International Conference on Mobile Data Management* 1 (2007), pp. 198–205. ISSN: 15516245. DOI: 10.1109/MDM.2007.36.

[94] P. Anantharam, C.a. Henson, K. Thirunarayan, and a.P. Sheth. « Trust model for semantic sensor and social networks: A preliminary report. » In: *Aerospace and Electronics Conference (NAECON), Proceedings of the IEEE 2010 National* (2010), pp. 1–5. ISSN: 0547-3578. DOI: 10.1109/NAECON.2010.5712915.

[95] Arvind Arasu, Shivnath Babu, and Jennifer Widom. « The CQL continuous query language: Semantic foundations and query execution. » In: *VLDB Journal* 15.2 (2006), pp. 121–142. ISSN: 10668888. DOI: 10.1007/s00778-004-0147-z.

[96] Narayanan C. Krishnan and Diane J. Cook. « Activity recognition on streaming sensor data. » In: *Pervasive and Mobile Computing* 10.164 (2014), pp. 138–154. ISSN: 15741192. DOI: 10.1016/j.pmcj.2012.07.003. URL: http://linkinghub.elsevier.com/retrieve/pii/S1574119212000776.

[97] Chun Wei Tsai, Chin Feng Lai, Ming Chao Chiang, and Laurence T. Yang. « Data mining for internet of things: A survey. » In: *IEEE Communications Surveys and Tutorials* 16.1 (2014), pp. 77–97. ISSN: 1553877X. DOI: 10.1109/SURV.2013.103013.00206.

[98] Maja Stikic, Diane Larlus, Sandra Ebert, and Bernt Schiele. « Weakly supervised recognition of daily life activities with wearable sensors. » In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33.12 (2011), pp. 2521–2537. ISSN: 01628828. DOI: 10.1109/TPAMI.2011.36.

[99] Burr Settles. *Active Learning*. Vol. 6. 1. 2012, pp. 1–114. ISBN: 9781608457250. DOI: 10.2200/S00429ED1V01Y201207AIM018.

[100] Muhammad Arshad Awan, Zheng Guangbin, and Shin Dug Kim. « A dynamic approach to recognize activities in WSN. » In: *International Journal of Distributed Sensor Networks* 2013 (2013). ISSN: 15501329. DOI: 10.1155/2013/385276.

[101] Roberto L. Shinmoto Torres, Damith C. Ranasinghe, Qinfeng Shi, and Anton van den Hengel. « Learning from Imbalanced Multiclass Sequential Data Streams Using Dynamically Weighted Conditional Random Fields. » In: (2016), pp. 1–28. arXiv: 1603.03627. URL: http://arxiv.org/abs/1603.03627.

[102] Darko Anicic, Sebastian Rudolph, Paul Fodor, and Nenad Stojanovic. « Stream reasoning and complex event processing in ETALIS. » In: *Semantic Web* 3.4 (2012), pp. 397–407. ISSN: 15700844. DOI: 10.3233/SW-2011-0053.

[103] Guus Schreiber and Yves Raimond. *RDF 1.1 Primer*. 2014. URL: http://www.w3.org/TR/rdf11-primer/.

[104] Dan Brickley and R.V. Guha. *RDF Schema 1.1 - W3C Recommendation*. 2008. DOI: 10.1016/B978-0-12-373556-0.00006-X. URL: http://www.w3.org/TR/rdf-schema/.

[105] P Hitzler Krötzsch, M., Parsia, B., Patel-Schneider, P.F., Rudolph, S. « OWL 2 Web Ontology Language Primer. » In: October (2009), pp. 1–46.

[106] Boris Motik et al. « OWL 2 Web Ontology Language - Structural Specification and Functional-Style Syntax (Second Edition). » In: *Online* December (2012), pp. 1–133. URL: https://www.w3.org/2007/OWL/draft/ED-owl2-syntax-20090914/all.pdf.

[107] Anastasia Analyti, Grigoris Antoniou, and Gerd Wagner. « Extended RDF as a Semantic Foundation of Rule Markup Languages. » In: *JAIR* 32 (2008), pp. 37–94. arXiv: arXiv:1111.0055v1.

[108] Andy Seaborne. *SPARQL 1.1 Property Paths*. 2010. URL: http://www.w3.org/TR/sparql11-property-paths/.

[109] Giuseppe De Giacomo and Maurizio Lenzerini. « TBox and ABox reasoning in expressive description logics. » In: 1 (1996).

[110] Ian Horrocks, Oliver Kutz, and Ulrike Sattler. « The Even More Irresistible SROIQ. » In: *Proc. of the 10th Int. Conf. on Principles of Knowledge Representation and Reasoning (KR2006)* (2006), pp. 57–67. URL: http://www.cs.man.ac.uk/{~}horrocks/Publications/download/2006/HoKS06a.pdf.

[111] Markus Krötzsch. « OWL 2 profiles: An introduction to lightweight ontology languages. » In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 7487 LNCS. 2012, pp. 112–183. ISBN: 9783642331572. DOI: 10.1007/978-3-642-33158-9_4.

[112] Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schneider. « Sweetening Ontologies with DOLCE. » In: *International Conference on Knowledge Engineering and Knowledge Management*. EKAW '02. London, UK, UK: Springer-Verlag, 2002, pp. 166–181. ISBN: 3-540-44268-5. URL: http://dl.acm.org/citation.cfm?id=645362.650863.

[113] B Smith and P Grenon. « Basic formal ontology (bfo). » In: *INFOMIS Reports* (2006). URL: http://scholar.google.com.br/scholar?hl=pt-BR{\&}q=Barry+Smith+Pierre+Grenon{\&}btnG={\&}lr={\#}5.

[114] Ian Niles and Adam Pease. « Towards a Standard Upper Ontology. » In: *The 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)* (2001), pp. 2–9. ISSN: 02632136. DOI: 10.1145/505168.505170. URL: http://doi.acm.org/10.1145/505168.505170{\%}5Cnhttp://dl.acm.org/ft{_}gateway.cfm?id=505170{\&}type=pdf.

[115] Werner Ceusters. « An information artifact ontology perspective on data collections and associated representational artifacts. » In: *Studies in Health Technology and Informatics*. Vol. 180. 2012, pp. 68–72. ISBN: 9781614991007. DOI: 10.3233/978-1-61499-101-4-68.

[116] H Boley, S Tabet, and G Wagner. « Design Rationale of RuleML: A Markup Language for Semantic Web Rules. » In: *The First International Semantic Web Working Symposium* 1 (2001), pp. 381–401. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.21.5035{\&}rep=rep1{\&}type=pdf.

[117] Mike Botts, George Percival, Carl Reed, and John Davidson. « OCG Sensor Web Enablement: Overview and High Level Architecture. » In: *International conference on GeoSensor Networks*. Springer, 2006, pp. 175–190. URL: http://www.jgrcs.info/index.php/jgrcs/article/view/276.

[118] Simon J D Cox. *Observations and Measurements - Part 1 - Observation schema*. 2007. DOI: http://www.opengeospatial.org/standards/om. URL: http://portal.opengeospatial.org/files/22466.

[119] M Botts and A Robin. « OpenGIS Sensor Model Language (SensorML) Implementation Specification. OGC 07-000. Open Geospatial Consortium, Inc., 180pp. » In: *portal. opengeospatial. org/files* (2007).

[120] Michael Compton, Cory Henson, Laurent Lefort, Holger Neuhaus, and Amit Sheth. « A survey of the semantic specification of sensors. » In: *CEUR Workshop Proceedings* 522 (2009), pp. 17–32. ISSN: 16130073. DOI: 10.1.1.352.9902.

[121] Krzysztof Janowicz and Michael Compton. « The Stimulus-Sensor-Observation Ontology Design Pattern and its Integration into the Semantic Sensor Network Ontology. » In: *Proceedings of the 3rd International Conference on Semantic Sensor Networks*. 2010, pp. 64–78. URL: https://www.geog.ucsb.edu/{~}jano/Semantic{_}Sensor{_}Ontology{_}2010.pdf.

[122] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. *From Data Mining to Knowledge Discovery in Databases*. 1996. DOI: 10.1609/aimag.v17i3.1230. arXiv: aimag.v17i3.1230. URL: http://www.aaai.org/ojs/index.php/aimagazine/article/view/1230.

[123] Lukasz a. Kurgan and Petr Musilek. « A survey of Knowledge Discovery and Data Mining process models. » In: *The Knowledge Engineering Review* 21.01 (2006), p. 1. ISSN: 0269-8889. DOI: 10.1017/S0269888906000737.

[124] Melanie Hilario, Phong Nguyen, Huyen Do, Adam Woznica, and Alexandros Kalousis. « Ontology-based meta-mining of knowledge discovery workflows. » In: *Studies in Computational Intelligence* 358 (2011), pp. 273–315. ISSN: 1860949X. DOI: 10.1007/978-3-642-20980-2_9.

[125] Floarea Serban, Joaquin Vanschoren, Jörg-Uwe Kietz, and Abraham Bernstein. « A survey of intelligent assistants for data analysis. » In: *ACM Computing Surveys* 45.3 (2013), pp. 1–35. ISSN: 03600300. DOI: 10.1145/2480741.2480748. URL: http://dl.acm.org/citation.cfm?id=2480748http://dl.acm.org/citation.cfm?doid=2480741.2480748.

[126] C. Maria Keet, Agnieszka Ławrynowicz, Claudia D'Amato, Alexandros Kalousis, Phong Nguyen, Raul Palma, Robert Stevens, and Melanie Hilario. « The Data Mining OPtimization Ontology. » In: *Web Semantics: Science, Services and Agents on the World Wide Web* 32 (2015), pp. 43–53. ISSN: 15708268. DOI: 10.1016/j.websem.2015.01.001. URL: http://linkinghub.elsevier.com/retrieve/pii/S1570826815000025.

[127] Panče Panov, Sašo Džeroski, and Larisa Soldatova. « OntoDM: An Ontology of Data Mining. » In: *2008 IEEE International Conference on Data Mining Workshops* (2008), pp. 752–760. DOI: 10.1109/ICDMW.2008.62. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4734003.

[128] Joaquin Vanschoren and Larisa N Soldatova. « Exposé: An ontology for data mining experiments. » In: *Proc. of the 3rd Int. Workshop on Third Generation Data Mining: Towards Service-oriented Knowledge Discovery (SoKD)* (2010), pp. 31–46.

[129] Panče Panov. « A modular ontology of data mining. » Doctoral Dissertation. Jozef Stefan International Postgraduate School, 2012, p. 191.

[130] Lynda Temal, Arnaud Rosier, Olivier Dameron, and Anita Burgun. « Mapping BFO and DOLCE. » In: *Studies in Health Technology and Informatics* 160.2 (2010), pp. 1065–1069. ISSN: 09269630. DOI: 10.3233/978-1-60750-588-4-1065.

[131] Panče Panov, Larisa Soldatova, and Sašo Džeroski. *Ontology of core data mining entities*. 2014, pp. 1222–1265. ISBN: 1061801403. DOI: 10.1007/s10618-014-0363-0. URL: http://link.springer.com/10.1007/s10618-014-0363-0.

[132] Panče Panov, Larisa N. Soldatova, and Sašo Džeroski. « Generic ontology of datatypes. » In: *Information Sciences* 329 (2016), pp. 900–920. ISSN: 00200255. DOI: 10.1016/j.ins.2015.08.006. URL: http://linkinghub.elsevier.com/retrieve/pii/S0020025515005800.

[133] Panče Panov, Larisa Soldatova, and Sašo Džeroski. « OntoDM-KDD: Ontology for representing the knowledge discovery process. » In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 8140 LNAI. Springer Berlin Heidelberg, 2013, pp. 126–140. ISBN: 9783642408960. DOI: 10.1007/978-3-642-40897-7_9.

[134] Douglas McIlwraith and Guang-Zhong Yang. « Body Sensor Networks for Sport, Wellbeing and Health. » In: *Signals and Communication Technology*. 2010, pp. 349–381. ISBN: 978-3-642-01340-9. DOI: 10.1007/978-3-642-01341-6_13. URL: http://www.springerlink.com/content/n716725j18626121/http://link.springer.com/10.1007/978-3-642-01341-6{_}13.

[135] Longbing Cao. « Behavior informatics: A new perspective. » In: *IEEE Intelligent Systems* 29.4 (2014), pp. 62–80. ISSN: 15411672. DOI: 10.1109/MIS.2014.60.

[136] Longbing Cao and Philip S. Yu. *Behavior computing: Modeling, analysis, mining and decision*. 2012, pp. 1–374. ISBN: 9781447129691. DOI: 10.1007/978-1-4471-2969-1.

[137] Chiara Renso, Miriam Baglioni, Jose António F de Macedo, Roberto Trasarti, and Monica Wachowicz. « How you move reveals who you are: Understanding human behavior by analyzing trajectory data. » In: *Knowledge and Information Systems* 37.2 (2013), pp. 331–362. ISSN: 02191377. DOI: 10.1007/s10115-012-0511-z.

[138] Longbing Cao, Philip S Yu, Hiroshi Motoda, and Graham Williams. « Special issue on behavior computing. » In: *Knowledge and Information Systems* 37.2 (2013), pp. 245–249. ISSN: 0219-1377. DOI: 10.1007/s10115-013-0668-0. URL: http://link.springer.com/10.1007/s10115-013-0668-0.

[139] G Ertek, Ayhan Demiriz, and Fatih Cakmak. « Linking behavioral patterns to personal attributes through data re-mining. » In: *Behavior Computing*. 2012. URL: http://link.springer.com/chapter/10.1007/978-1-4471-2969-1{_}12.

[140] Belkacem Chikhaoui, Shengrui Wang, Tengke Xiong, and Hélène Pigot. « Pattern-based causal relationships discovery from event sequences for modeling behavioral user profile in ubiquitous environments. » In: *Information Sciences* 285 (2014), pp. 204–222. ISSN: 00200255. DOI: 10.1016/j.ins.2014.06.026. URL: http://linkinghub.elsevier.com/retrieve/pii/S0020025514006549.

[141] Leo Sauermann. « The Gnowsis Semantic Desktop approach to Personal Information Management. » Doctoral Dissertation. Universität Kaiserslautern, 2009, p. 294. ISBN: 9783866244498. URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.172.5430{\&}rep=rep1{\&}type=pdf.

[142] Rania Khéfifi. « Informations personnelles sensibles aux contextes : modélisation, interrogation et composition. » PhD thesis. Université Paris Sud - Paris XI, 2014, p. 180.

[143] Benjamin C. M. Fung, Ke Wang, Rui Chen, and Philip S. Yu. « Privacy-preserving data publishing: A Survey of Recent Developments. » In: *CSUR* 42.4 (2010), pp. 1–53. ISSN: 03600300. DOI: 10.1145/1749603.1749605. URL: http://portal.acm.org/citation.cfm?doid=1749603.1749605.

[144] Ann Cavoukian and Khaled El Emam. « De-identification Protocols: » in: (2014).

[145] Anna Monreale, Salvatore Rinzivillo, Francesca Pratesi, Fosca Giannotti, and Dino Pedreschi. « Privacy-by-design in big data analytics and social mining. » In: *EPJ Data Science* 3.1 (2014), p. 10. DOI: 10.1140/epjds/s13688-014-0010-4. URL: http://dx.doi.org/10.1140/epjds/s13688-014-0010-4.

[146] Khaled El Emam. *Guide to the De-Identification of Personal Health Information*. CRC Press, 2013. ISBN: 9781439809822.

[147] Simson Garfinkel. *De-Identification of Personal Information*. Tech. rep. Gaithersburg, MD: National Institute of Standards and Technology, Information Access Division, Information Technology Laboratory, 2015, p. 46. DOI: 10.6028/NIST.IR.8053.

[148] Zhiqiang Yang and Rebecca N. Wright. « Privacy-preserving computation of bayesian networks on vertically partitioned data. » In: *IEEE Transactions on Knowledge and Data Engineering* 18.9 (2006), pp. 1253–1264. ISSN: 10414347. DOI: 10.1109/TKDE.2006.147.

[149] Jaideep Vaidya and Chris Clifton. « Privacy-Preserving K-Means Clustering over Vertically Partitioned Data. » In: *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining* (2003), pp. 206–215. DOI: 10.1145/956750.956776.

[150] Justin Zhan and Stan Matwin. « Privacy-preserving support vector machine classification. » In: *International Journal of Intelligent Information and Database Systems* 1.3/4 (2007), p. 356. ISSN: 1751-5858. DOI: 10.1504/IJIIDS.2007.016686. URL: http://www.inderscience.com/link.php?id=16686.

[151] Jaideep Vaidya, Chris Clifton, Murat Kantarcioglu, and a. Scott Patterson. « Privacy-preserving decision trees over vertically partitioned data. » In: *ACM Transactions on Knowledge Discovery from Data* 2.3 (2008), pp. 1–27. ISSN: 15564681. DOI: 10.1145/1409620.1409624.

[152] Stan Matwin. « Privacy-Preserving Data Mining Techniques: Survey and Challenges. » In: *Discrimination & Privacy in the Information Society*. 2013, pp. 209–221.

[153] Nancy Victor, Daphne Lopez, and Jemal H Abawajy. « Privacy models for big data: a survey. » In: *International Journal of Big Data Intelligence* 3.1 (2016), p. 61. ISSN: 2053-1389. DOI: 10.1504/IJBDI.2016.073904. URL: http://www.inderscience.com/link.php?id=73904.

[154] Indrajit Roy, S.T.V. Srinath T V Setty, Ann Kilzer, Vitaly Shmatikov, and Emmett Witchel. « Airavat: Security and privacy for MapReduce. » In: *Proceedings of the 7th USENIX conference on Networked systems design and implementation* (2010), pp. 20–20. DOI: 10.1.1.149.2533. URL: http://dl.acm.org/citation.cfm?id=1855731.

[155] X Zhang, L T Yang, C Liu, and J Chen. « A Scalable Two-Phase Top-Down Specialization Approach for Data Anonymization Using MapReduce on Cloud. » In: *IEEE Transactions on Parallel and Distributed Systems* 25.2 (2014), pp. 363–373. ISSN: 1045-9219. DOI: 10.1109/TPDS.2013.48.

[156] Jianneng Cao, Barbara Carminati, Elena Ferrari, and Kian Lee Tan. « CASTLE: Continuously anonymizing data streams. » In: *IEEE Transactions on Dependable and Secure Computing* 8.3 (2011), pp. 337–352. ISSN: 15455971. DOI: 10.1109/TDSC.2009.47.

[157] J Cao, P Karras, P Kalnis, and KL Tan. « SABRE: a Sensitive Attribute Bucketization and REdistribution framework for t-closeness. » In: *VLDB* (2011), pp. 59–81. DOI: 10.1007/s00778-010-0191-9. URL: http://link.springer.com/article/10.1007/s00778-010-0191-9.

[158] Maumita Bhattacharya, Rafiqul Islam, and Jemal Abawajy. « Evolutionary optimization: A big data perspective. » In: *Journal of Network and Computer Applications* 59 (2014), pp. 416–426. ISSN: 10848045. DOI: 10.1016/j.jnca.2014.07.032. URL: http://linkinghub.elsevier.com/retrieve/pii/S1084804514001805.

[159] Dai Hai Ton That, Iulian Sandu Popa, Karine Zeitouni, and Cristian Borcea. « PAMPAS. » In: *Proceedings of the 28th International Conference on Scientific and Statistical Database Management - SSDBM '16*. New York, New York, USA: ACM Press, 2016, pp. 1–12. ISBN: 9781450342155. DOI: 10.1145/2949689.2949704. URL: http://dl.acm.org/citation.cfm?doid=2949689.2949704.

[160] Ravi S Sandhu, Edward J Coyne, Hal L Feinstein, and Charles E Youman. « Role-Based Access Control Models. » In: *IEEE Computer* 29.2 (1995), pp. 38–47. ISSN: 00189162. DOI: 10.1109/2.485845.

[161] D R Kuhn, E J Coyne, and T R Weil. « Adding Attributes to Role Based Access Control. » In: *Computer* 43.6 (2010), pp. 79–81.

[162] Vincent C. Hu, D. Richard Kuhn, and David F. Ferraiolo. « Attribute-based access control. » In: *Computer* 48.2 (2015), pp. 85–88. ISSN: 00189162. DOI: 10.1109/MC.2015.33.

[163] Alan H Karp, Harry Haury, and Michael H Davis. « From ABAC to ZBAC : The Evolution of Access Control Models. » In: *ISSA Journal* April (2010), pp. 22–30.

[164] Marc Langheinrich. « A Privacy Awareness System for Ubiquitous Computing Environments. » In: *UbiComp '02: Proceedings of the 4th international conference on Ubiquitous Computing*. 2002, pp. 237–245. ISBN: 9783540442677. DOI: 10.1007/3-540-45809-3_19. URL: http://link.springer.com/10.1007/3-540-45809-3{_}19.

[165] Zhang Qingsheng, Qi Yong, Zhao Jizhong, Hou Di, Zhao Tianhai, and Liu Liang. « A study on context-aware privacy protection for personal information. » In: *Proceedings - International Conference on Computer Communications and Networks, ICCCN*. 2007, pp. 1351–1358. ISBN: 9781424412518. DOI: 10.1109/ICCCN.2007.4318009.

[166] C. A. Ardagna, M. Cremonini, S. De Capitani Di Vimercati, and P. Samarati. « A privacy-aware access control system. » In: *Journal of Computer Security* 16.4 (2008), pp. 369–397. ISSN: 0926227X. DOI: 10.3233/JCS-2008-0328.

[167] Wassim Itani, Ayman Kayssi, and Ali Chehab. « Privacy as a service: Privacy-aware data storage and processing in cloud computing architectures. » In: *8th IEEE International Symposium on Dependable, Autonomic and Secure Computing, DASC 2009*. 2009, pp. 711–716. ISBN: 9780769539294. DOI: 10.1109/DASC.2009.139.

[168] Martin Kost and JC Freytag. « Privacy Analysis using Ontologies. » In: *... on Data and Application Security and Privacy* (2012), pp. 205–216. URL: http://dl.acm.org/citation.cfm?id=2133627.

[169] Supriyo Chakraborty, Chenguang Shen, Kasturi Rangan Raghavan, Yasser Shoukry, Matt Miller, and Mani B. Srivastava. « ipShield : A Framework For Enforcing Context-Aware Privacy. » In: *Nsdi'14* (2014), pp. 143–156.

[170] Evangelos Pournaras, Izabela Moise, and Dirk Helbing. « Privacy-preserving ubiquitous social mining via modular and compositional virtual sensors. » In: *Proceedings - International Conference on Advanced Information Networking and Applications, AINA* 2015-April.October 2014 (2015), pp. 332–338. ISSN: 1550445X. DOI: 10.1109/AINA.2015.203.

[171] Torsten Priebe, Wolfgang Dobmeier, and Nora Kamprath. « Supporting attribute-based access control with ontologies. » In: *Proceedings - First International Conference on Availability, Reliability and Security, ARES 2006* 2006 (2006), pp. 465–472. DOI: 10.1109/ARES.2006.127.

[172] Kheira Bekara, Yosra Ben Mustapha, and Maryline Laurent. « XPACML eXtensible Privacy Access Control Markup Langua. » In: *The Second International Conference on Communications and Networking*. IEEE, 2010, pp. 1–5. ISBN: 978-1-4244-8839-1. DOI: 10.1109/COMNET.2010.5699807. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5699807.

[173] Amit Sheth, Pramod Anantharam, and Cory Henson. « Semantic, cognitive, and perceptual computing: Paradigms that shape human experience. » In: *Computer* 49.3 (2016), pp. 64–72. ISSN: 00189162. DOI: 10.1109/MC.2016.75.

[174] Natalia Díaz Rodríguez, M. P. Cuéllar, Johan Lilius, and Miguel Delgado Calvo-Flores. « A survey on ontologies for human behavior recognition. » In: *ACM Computing Surveys* 46.4 (2014), pp. 1–33. ISSN: 03600300. DOI: 10.1145/2523819. URL: http://dl.acm.org/citation.cfm?doid=2597757.2523819.

[175] Barry Smith et al. « The OBO Foundry: coordinated evolution of ontologies to support biomedical data integration. » In: *Nat Biotech* 25.11 (2007), pp. 1251–1255. ISSN: 1087-0156. URL: http://dx.doi.org/10.1038/nbt1346.

[176] Konstantinos Kotis and George A. Vouros. « Human-centered ontology engineering: The HCOME methodology. » In: *Knowledge and Information Systems* 10.1 (2006), pp. 109–131. ISSN: 0219-1377. DOI: 10.1007/s10115-005-0227-4. URL: http://link.springer.com/10.1007/s10115-005-0227-4.

[177] Claudio Masolo, Stefano Borgo, Aldo Gangemi, Nicola Guarino, and Alessandro Oltramari. « WonderWeb Deliverable D18. » In: *Communities* 2003 (2003), p. 343. ISSN: 00083100. URL: http://wonderweb.semanticweb.org/deliverables/documents/D18.pdf.

[178] Domenico Pisanelli, Aldo Gangemi, and Steve Geri. « An ontology of descriptions and situations for Lyee's hypothetical world. » In: *Proceedings Somet Workshop* (2003).

[179] Viviana Mascardi, Valentina Cordì, and Paolo Rosso. « A Comparison of Upper Ontologies. » In: *Woa* (2007), pp. 55–64. DOI: 10.1.1.107.1689.

[180] Pier Buttigieg, Norman Morrison, Barry Smith, Christopher J Mungall, and Suzanna E Lewis. « The environment ontology: contextualising biological and biomedical entities. » In: *Journal of Biomedical Semantics* 4.1 (2013), p. 43. ISSN: 2041-1480. DOI: 10.1186/2041-1480-4-43. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3904460{\&}tool=pmcentrez{\&}rendertype=abstract{\%}5Cnhttp://www.jbiomedsem.com/content/4/1/43.

[181] RH Scheuermann, W Ceusters, and B Smith. « Toward an Ontological Treatment of Disease and Diagnosis. » In: *Proceedings of the Second AMIA Summit on Translational Bioinformatics* September 2016 (2009), pp. 116–120.

[182] Janna Hastings and Werner Ceusters. « Representing mental functioning: Ontologies for mental health and disease. » In: *ICBO 2012: 3rd International Conference on Biomedical Ontology* (2012), pp. 1–5. URL: http://ontology.buffalo.edu/smith/articles/ICBO2012/MFO{_}Hastings.pdf.

[183] Georgios V. Gkoutos, Paul N. Schofield, and Robert Hoehndorf. *The Neurobehavior Ontology. An Ontology for Annotation and Integration of Behavior and Behavioral Phenotypes.* 1st ed. Vol. 103. Elsevier Inc., 2012, pp. 69–87. ISBN: 9780123884084. DOI: 10.1016/B978-0-12-388408-4.00004-6. URL: http://dx.doi.org/10.1016/B978-0-12-388408-4.00004-6.

[184] Georgios V Gkoutos, Eain C J Green, Ann-marie Mallon, John M Hancock, and Duncan Davidson. « Using ontologies to describe mouse phenotypes. » In: *Genome biology* 6.1 (2005), R8. ISSN: 1474-760X. DOI: 10.1186/gb-2004-6-1-r8. URL: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=549069{\&}tool=pmcentrez{\&}rendertype=abstracthttp://genomebiology.biomedcentral.com/articles/10.1186/gb-2004-6-1-r8http://www.ncbi.nlm.nih.gov/pubmed/15642100http://www.pubmedcentral.nih.gov/article.

[185] J. A. Blake et al. « Gene Ontology Annotations and Resources. » In: *Nucleic Acids Research* 41.D1 (2013), pp. D530–D535. ISSN: 0305-1048. DOI: 10.1093/nar/gks1050. URL: http://nar.oxfordjournals.org/lookup/doi/10.1093/nar/gks1050https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gks1050.

[186] Mélanie Courtot, Frank Gibson, Allyson L. Lister, James Malone, Daniel Schober, Ryan R. Brinkman, and Alan Ruttenberg. « MIREOT: The minimum information to reference an external ontology term. » In: *Applied Ontology* 6.1 (2011), pp. 23–33. ISSN: 15705838. DOI: 10.3233/AO-2011-0087.

[187] Zuoshuang Xiang, Mélanie Courtot, Ryan R Brinkman, Alan Ruttenberg, and Yongqun He. « OntoFox: web-based support for ontology reuse. » In: *BMC research notes* 3.175 (2010), pp. 1–12. ISSN: 1756-0500. DOI: 10.1186/1756-0500-3-175.

[188] William Jones, Harry Bruce, Marcia J. Bates, Nicholas Belkin, Ofer Bergman, and Cathy Marshall. « Personal information management in the present and future perfect: Reports from a special NSF-sponsored workshop. » In: *Proceedings of the American Society for Information Science and Technology* 42.1 (2006), pp. 1–67. ISSN: 00447870. DOI: 10.1002/meet.1450420151. URL: http://doi.wiley.com/10.1002/meet.1450420151.

[189] Vivi Katifori, Antonella Poggi, Monica Scannapieco, Tiziana Catarci, and Yannis Ioannidis. « OntoPIM: How to rely on a personal ontology for personal information management. » In: *CEUR Workshop Proceedings* 175.i (2005), pp. 2–6. ISSN: 16130073.

[190] Eleni Kargioti, Efstratios Kontopoulos, and Nick Bassiliades. « OntoLife: An ontology for semantically managing personal information. » In: *IFIP International Federation for Information Processing* 296 (2009), pp. 127–133. ISSN: 15715736. DOI: 10.1007/978-1-4419-0221-4_16.

[191] R Luca, S I Bejinariu, and H. N. Teodorescu. « An ontology of human walk for autonomous systems. » In: *18th International Conference on System Theory, Control and Computing, ICSTCC 2014*. 2014, pp. 488–493. ISBN: 9781479946013. DOI: 10.1109/ICSTCC.2014.6982464.

[192] Qin Ni, Iván Pau De La Cruz, and Ana Belén García Hernando. « A foundational ontology-based model for human activity representation in smart homes. » In: *Journal of Ambient Intelligence and Smart Environments* 8.1 (2016), pp. 47–61. ISSN: 18761364. DOI: 10.3233/AIS-150359.

[193] Davy Preuveneers, Jan Van Den Bergh, Dennis Wagelaar, Andy Georges, Peter Rigole, Tim Clerckx, Yolande Berbers, Karin Coninx, Viviane Jonckers, and Koen De Bosschere. « Towards an Extensible Context Ontology for Ambient Intelligence. » In: *Ambient Intelligence* 3295 (2004), pp. 148–159. ISSN: 03029743. DOI: 10.1007/978-3-540-30473-9_15. arXiv:

1008.1900. URL: http://link.springer.com/chapter/10.1007/978-3-540-30473-9{_}15.

[194] Ramón Hervás, José Bravo, and Jesús Fontecha. « A Context Model based on Ontological Languages: a Proposal for Information Visualization. » In: *J. Ucs* 16.12 (2010), pp. 1539–1555. ISSN: 09486968. DOI: 10.3217/jucs-016-12-1539.

[195] Daniele Riboni and Claudio Bettini. « OWL 2 modeling and reasoning with complex human activities. » In: *Pervasive and Mobile Computing* 7.3 (2011), pp. 379–395. ISSN: 15741192. DOI: 10.1016/j.pmcj.2011.02.001.

[196] a Katifori and C Vassilakis. « Using spreading activation through ontologies to support personal information management. » In: *Proc. of Common Sense Knowledge* (2008). URL: http://ceur-ws.org/Vol-323/paper05.pdf.

[197] Louis Atallah and Guang-Zhong Yang. « The use of pervasive sensing for behaviour profiling — a survey. » In: *Pervasive and Mobile Computing* 5.5 (2009), pp. 447–464. ISSN: 15741192. DOI: 10.1016/j.pmcj.2009.06.009. URL: http://linkinghub.elsevier.com/retrieve/pii/S1574119209000583.

[198] J. D. Álvarez, J. L. Redondo, E. Camponogara, J. Normey-Rico, M. Berenguel, and P. M. Ortigosa. « Optimizing building comfort temperature regulation via model predictive control. » In: *Energy and Buildings* 57 (2013), pp. 361–372. ISSN: 03787788. DOI: 10.1016/j.enbuild.2012.10.044. URL: http://dx.doi.org/10.1016/j.enbuild.2012.10.044.

[199] Miguel Molina-Solana, María Ros, M. Dolores Ruiz, Juan Gómez-Romero, and María J. Martín-Bautista. « Data Science for Building Energy Management: a Review. » In: *Renewable and Sustainable Energy Reviews* 70.May 2016 (2017), pp. 598–609. ISSN: 13640321. DOI: 10.1016/j.rser.2016.11.132. URL: http://dx.doi.org/10.1016/j.rser.2016.11.132.

[200] Cory Henson, Krishnaprasad Thirunarayan, and Amit Sheth. « An ontological approach to focusing attention and enhancing machine perception on the Web. » In: *Applied Ontology* 6.4 (2011), pp. 345–376. ISSN: 15705838. DOI: 10.3233/AO-2011-0100.

[201] Huiyong Xiao and Isabel F. Cruz. « A multi-ontology approach for personal information management. » In: *CEUR Workshop Proceedings* 175 (2005). ISSN: 16130073.

[202] Stephan Sigg, Dawud Gordon, Georg Von Zengen, Michael Beigl, Sandra Haseloff, and Klaus David. « Investigation of context prediction accuracy for different context abstraction levels. » In: *IEEE Transactions on Mobile Computing* 11.6 (2012), pp. 1047–1059. ISSN: 15361233. DOI: 10.1109/TMC.2011.170.

[203] Frieder Ganz, Daniel Puschmann, Payam Barnaghi, and Francois Carrez. « A Practical Evaluation of Information Processing and Abstraction Techniques for the Internet of Things. » In: *IEEE Internet of Things Journal* 2.4 (2015), pp. 340–354. ISSN: 23274662. DOI: 10.1109/JIOT.2015.2411227.

[204] Cristiano Castelfranchi and Fabio Paglieri. « The role of beliefs in goal dynamics: prolegomena to a constructive theory of intentions. » In: *Synthese* 155.2 (2007), pp. 237–263. ISSN: 0039-7857. DOI: 10.1007/s11229-006-9156-3. URL: http://dx.doi.org/10.1007/s11229-006-9156-3http://link.springer.com/10.1007/s11229-006-9156-3.

[205] Li Liu, Yuxin Peng, Ming Liu, and Zigang Huang. « Sensor-based human activity recognition system with a multilayered model using time series shapelets. » In: *Knowledge-Based Systems* 90 (2015), pp. 138–152. ISSN: 09507051. DOI: 10.1016/j.knosys.2015.09.024. URL: http://linkinghub.elsevier.com/retrieve/pii/S0950705115003639.

[206] Abdur Rahim Mohammad Forkan, Ibrahim Khalil, and Mohammed Atiquzzaman. « ViSiBiD: A learning model for early discovery and real-time prediction of severe clinical events using Vital Signs as Big Data. » In: *Computer Networks* 113 (2017), pp. 244–257. ISSN: 13891286. DOI: 10.1016/j.comnet.2016.12.019. URL: http://linkinghub.elsevier.com/retrieve/pii/S1389128616304431.

[207] Barbara Furletti, Paolo Cintia, Chiara Renso, and Laura Spinsanti. « Inferring human activities from GPS tracks. » In: *Proceedings of the 2nd ACM SIGKDD International Workshop on Urban Computing - UrbComp '13*. August 2015. New York, New York, USA: ACM, 2013, p. 1. ISBN: 9781450323314. DOI: 10.1145/2505821.2505830. URL: http://dl.acm.org/citation.cfm?doid=2505821.2505830.

[208] Markus Krötzsch, F Simancik, and Ian Horrocks. « A description logic primer. » In: *arXiv preprint arXiv:1201.4089* June (2012), pp. 1–17. arXiv: 1201.4089. URL: http://arxiv.org/abs/1201.4089.

[209] Steve Harris and Andy Seaborne. *SPARQL 1.1 Query Language*. 2013. DOI: citeulike-article-id:2620569. URL: http://www.w3.org/TR/2013/REC-sparql11-query-20130321/.

[210] Evren Sirin and Bijan Parsia. « SPARQL-DL: SPARQL Query for OWL-DL. » In: *n 3rd OWL Experiences and Directions Workshop (OWLED-2007)*. 2007, pp. 1–10.

[211] Egor V. Kostylev, Juan L. Reutter, Miguel Romero, and Domagoj Vrgoč. « SPARQL with property paths. » In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 9366 (2015), pp. 3–18. ISSN: 16113349. DOI: 10.1007/978-3-319-25007-6_1.

[212] Oasis. « eXtensible Access Control Markup Language Version 3.0. » In: *OASIS Standard* February (2013), p. 154.

[213] Fatemeh Asadi, Massimiliano Di Penta, and Giuliano Antoniol. « A Heuristic-based Approach to Identify Concepts in Execution Traces. » In: ().

[214] Man Ching Yuen, Irwin King, and Kwong Sak Leung. « A survey of crowdsourcing systems. » In: *Proceedings - 2011 IEEE International Conference on Privacy, Security, Risk and Trust and IEEE International Conference on Social Computing, PASSAT/SocialCom 2011* (2011), pp. 766–773. DOI: 10.1109/PASSAT/SocialCom.2011.36.

[215] Sébastien Harispe, Sylvie Ranwez, Stefan Janaqi, and Jacky Montmain. « Semantic Similarity from Natural Language and Ontology Analysis. » In: *Synthesis Lectures on Human Language Technologies* 8.1 (2015), pp. 1–254. ISSN: 1947-4040. DOI: 10.2200/S00639ED1V01Y201504HLT027.

[216] Payam Barnaghi, Wei Wang, Lijun Dong, and Chonggang Wang. « A linked-data model for semantic sensor streams. » In: *Proceedings - 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, GreenCom-iThings-CPSCom 2013* (2013), pp. 468–475. DOI: 10.1109/GreenCom-iThings-CPSCom.2013.95.

[217] Z.a b Chen and N.a b Chen. « Provenance information representation and tracking for remote sensing observations in a Sensor Web enabled environment. » In: *Remote Sensing* 7.6 (2015), pp. 7646–7670. ISSN: 2072-4292. DOI: 10.3390/rs70607646. URL: http://www.scopus.com/inward/record.url?eid=2-s2.0-84933574049{\&}partnerID=40{\&}md5=1fdc7c948a0ea3054f279b4a193b142a.

[218] Charith Perera, Arkady Zaslavsky, Chi Harold Liu, Michael Compton, Peter Christen, and Dimitrios Georgakopoulos. « Sensor Search Techniques for Sensing as a Service Architecture for the Internet of Things. » In: *IEEE Sensors Journal* 14.2 (2014), pp. 406–420. ISSN: 1530-437X. DOI: 10.1109/JSEN.2013.2282292. arXiv: arXiv:1309.3618v1. URL: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6605518.

[219] Sharwari Satish Solapure and Harish Kenchannavar. « Internet of Things: A survey related to various recent architectures and platforms available. » In: *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. C. IEEE, 2016, pp. 2296–2301. ISBN: 978-1-5090-2029-4. DOI: 10.1109/ICACCI.2016.7732395. URL: http://ieeexplore.ieee.org/document/7732395/.

[220] Karl Aberer, Manfred Hauswirth, and Ali Salehi. *The Global Sensor Networks middleware for efficient and flexible deployment and interconnection of sensor networks*. 2006.

[221] Radoslaw Oldakowski and Christian Bizer. « SemMF : A Framework for Calculating Semantic Similarity of Objects Represented as RDF Graphs. » In: *Poster at the 4th International Semantic Web Conference (ISWC 2005)* October (2005), pp. 1–3.

[222] Thiago Moreira da Costa, Elie Rachkidi, Nazim Agoulmine, and Hervé Martin. « An experiment on deploying a privacy-aware sensing as a service in the sensor cloud. » In: *IEEE Advanced*. Paris: IEEE Computer Society, 2017.

[223] Mihai Codescu and Gregor Horsinka. « Osmonto-an ontology of openstreetmap tags. » In: *Fourth International Conference on GeoSpatial Semantics (GeoS-11)*. 2010, pp. 1–10. URL: http://www.inf.unibz.it/{~}okutz/resources/osmonto.pdf.

[224] Mohamed Morsey, Jens Lehmann, Sören Auer, and Axel Cyrille Ngonga Ngomo. « DBpedia SPARQL benchmark - Performance assessment with real queries on real data. » In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Vol. 7031 LNCS. PART 1. 2011, pp. 454–469. ISBN: 9783642250729. DOI: 10.1007/978-3-642-25073-6_29.

[225] Ilianna Kollia and Birte Glimm. « Optimizing SPARQL query answering over OWL ontologies. » In: *Journal of Artificial Intelligence Research* 48 (2013), pp. 253–303. ISSN: 10769757. DOI: 10.1613/jair.3872. arXiv: 1402.0576.

[226] Christian Bizer and Andreas Schultz. « The Berlin SPARQL Benchmark. » In: *International Journal on Semantic Web and Information Systems* 5.2 (2001), pp. 1–24. ISSN: 1552-6283. DOI: 10.4018/jswis.2009040101.

[227] Lei Zou, M. Tamer Özsu, Lei Chen, Xuchuan Shen, Ruizhe Huang, and Dongyan Zhao. « gStore: A graph-based SPARQL query engine. » In: *VLDB Journal* 23.4 (2014), pp. 565–590. ISSN: 0949877X. DOI: 10.1007/s00778-013-0337-7.

[228] Jin-cui YANG and Bin-xing FANG. « Security model and key technologies for the Internet of things. » In: *The Journal of China Universities of Posts and Telecommunications* 18 (2011), pp. 109–112. ISSN: 10058885. DOI: 10.1016/S1005-8885(10)60159-8. URL: http://www.sciencedirect.com/science/article/pii/S1005888510601598.

[229] Nikolaos Papailiou, Dimitrios Tsoumakos, Panagiotis Karras, and Nectarios Koziris. « Graph-Aware , Workload-Adaptive SPARQL Query Caching. » In: *Sigmod* (2015), pp. 1777–1792. ISSN: 07308078. DOI: 10.1145/2723372.2723714.

[230] Paolo Missier, Khalid Belhajjame, and James Cheney. « The W3C PROV family of specifications for modelling provenance metadata. » In: *Proceedings of the 16th International Conference on Extending Database Technology - EDBT '13* (2013), p. 773. DOI: 10.1145/2452376.2452478. URL: http://dl.acm.org/citation.cfm?doid=2452376.2452478.

[231] Larisa N. Soldatova, Wayne Aubrey, Ross D. King, and Amanda Clare. « The EXACT description of biomedical protocols. » In: *Bioinformatics* 24.13 (2008). ISSN: 13674803. DOI: 10.1093/bioinformatics/btn156. URL: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2718634/.