A Semantic Sensor Web Framework for Proactive Environmental Monitoring and Control

by

JUDE ADEKUNLE ADELEKE

Submitted in fulfillment of the academic requirements for the degree of
Doctor of Philosophy
in the
School of Mathematics, Statistics & Computer Science,
University of KwaZulu-Natal,
Durban, South Africa.

June 2017

As the candidate's supervisors, we have approved this dissertation for submission

Signed:.................................. Name:... Date:............

Signed:.................................. Name:... Date:............

Signed:.................................. Name:... Date:............

Abstract

Observing and monitoring of the natural and built environments is crucial for maintaining and preserving human life. Environmental monitoring applications typically incorporate some sensor technology to continually observe specific features of interest in the physical environment and transmitting data emanating from these sensors to a computing system for analysis. Semantic Sensor Web technology supports semantic enrichment of sensor data and provides expressive analytic techniques for data fusion, situation detection and situation analysis.

Despite the promising successes of the Semantic Sensor Web technology, current Semantic Sensor Web frameworks are typically focused at developing applications for detecting and reacting to situations detected from current or past observations. While these reactive applications provide a quick response to detected situations to minimize adverse effects, they are limited when it comes to anticipating future adverse situations and determining proactive control actions to prevent or mitigate these situations. Most current Semantic Sensor Web frameworks lack two essential mechanisms required to achieve proactive control, namely, mechanisms for anticipating the future and coherent mechanisms for consistent decision processing and planning.

Designing and developing proactive monitoring and control Semantic Sensor Web applications is challenging. It requires incorporating and integrating different techniques for supporting situation detection, situation prediction, decision making and planning in a coherent framework. This research proposes a coherent Semantic Sensor Web framework for proactive monitoring and control. It incorporates ontology to facilitate situation detection from streaming sensor observations, statistical machine learning for situation prediction and Markov Decision Processes for decision making and planning. The efficacy and use of the framework is evaluated through the development of two different prototype applications. The first application is for proactive monitoring and control of indoor air quality to avoid poor air quality situations. The second is for proactive monitoring and control of electricity usage in blocks of residential houses to prevent strain on the national grid. These applications show the effectiveness of the proposed framework for developing Semantic Sensor Web applications that proactively avert unwanted environmental situations before they occur.

Preface

The research described in this dissertation was carried out in the School of Mathematics, Statistics and Computer Science, University of KwaZulu-Natal, Durban, from July 2013 to June 2017, under the supervision of Prof. Deshendran Moodley, Dr. Gavin Rens and Prof. Aderemi Oluyinka Adewumi.

These studies represent original work by the author and have not otherwise been submitted in any form for any degree or diploma to any tertiary institution. Where use has been made of the work of others it is duly acknowledged in the text.

UNIVERSITY OF KWAZULU-NATAL
COLLEGE OF AGRICULTURE, ENGINEERING AND SCIENCE

Declaration 1 - Plagiarism

I, Jude Adekunle Adeleke declare that

1. The research reported in this thesis, except where otherwise indicated, is my original research.

2. This thesis has not been submitted for any degree or examination at any other university

3. This thesis does not contain other persons' data, pictures, graphs or other information, unless specifically acknowledged as being sourced from other persons.

4. This thesis does not contain other persons' writing, unless specifically acknowledged as being sourced from other researchers. Where other written sources have been quoted, then:

   (a) Their words have been re-written but the general information attributed to them has been referenced.

   (b) Where their exact words have been used, then their writing has been placed in italics and inside quotation marks, and referenced.

5. This thesis does not contain texts, graphics or tables copied and pasted from the Internet, unless specifically acknowledged, and the source being detailed in the thesis and in the References sections.

Signed:..

UNIVERSITY OF KWAZULU-NATAL
COLLEGE OF AGRICULTURE, ENGINEERING AND SCIENCE

Declaration 2 -Publications

1. **Adeleke, J.A. and Moodley, D.** (2015), September. An Ontology for Proactive Indoor Environmental Quality Monitoring and Control. In *Proceedings of the 2015 Annual Research Conference on South African Institute of Computer Scientists and Information Technologists (p. 2).* ACM, New York, NY, USA. doi:10.1145/2815782.2815816

2. **Adeleke, J.A, Moodley D., Rens, G. and Adewumi, A.O.** (2017). Integrating Statistical Machine Learning in a Semantic Sensor Web for Proactive Monitoring and Control. *Sensors*, 17(4), 807; doi:10.3390/s17040807

Signed:..

Dedication

To God, the Almighty,
the epitome of love and wisdom,
your supreme intelligence is evident in the creatures of the universe,
some of which we attempt to emulate
and integrate in artificial
artifacts.

And

the memory of my late parents,
Pa. Joseph O. Adeleke and Mrs. Sarah N. Adeleke,
your passion for education, and your role model in diligence and humanity
has motivated me thus far. I only wish you were around
to be proud of this achievement of yours.
You will always be in my memory.

Acknowledgments

First and foremost, my profound gratitude goes to the Almighty God Jehovah, and my Lord Jesus Christ, for the gift of life, the grace and immeasurable blessings that have kept me through the journey of life, including the high and low periods of this work. You are the *Alpha* and the *Omega*, the maker of the heavens, the earth and all things therein. May your name be praised from time indefinite to time indefinite, even forever. *Amen!*

I am very grateful to my supervisor, Prof. Deshendran Moodley, for steering the course of this research, for his immeasurable support and for affording me the opportunity of funding through the Center for Artificial Intelligence Research (CAIR). I am also very grateful for the invaluable support of my co-supervisors Prof. Aderemi O. Adewumi and Dr. Gavin Rens. Without your support and guidance, this work would not have been possible.

Many thanks to the entire faculty and members of the Adaptive Systems Laboratory, the CAIR node in UKZN. Anban, Edgar and Sue have been sources of invaluable support and encouragements. The feedbacks in the group meetings have been very helpful in this research. I acknowledge the invaluable support of Prof. Naidoo and members of his occupational health team - Jafta and Busisiwe during field work on the cohort study.

I like to acknowledge the visionary leadership of the Director General and the entire management of the National Space Research and Development Agency (NASRDA) of Nigeria, for granting me a study leave and an extension to the leave for the successful completion of this work.

So many people have played invaluable roles along the journey of this research. I may not be able to mention all your names but I appreciate you all. I like to thank Laila for effective administrative support, Soren for technical support, Charmaine for tirelessly taking delivery of sensors and equipments, and Rolan for tirelessly proofreading parts of this thesis.

This acknowledgment will not be complete without thanking Maria Keet, who originally accepted me as a PhD student, and more so for her sheer generosity for finding me a capable hand to continue with the supervision of my PhD prior to her departure from UKZN.

Contents

List of Figures

List of Tables

Glossary

ANN	Artificial Neural Networks
AQI	Air Quality Index
ARIMA	Autoregressive Integrated Moving Average
ARMA	AutoRegressive Moving Average
BN	Bayesian Networks
CISRO	Commonwealth Scientific and Industrial Research Organisation
CQ	Competence Question
DBMS	Database Management System
DL	Descrptive Language
DSM	Demand Side Management
DSMS	Data Streams Management System
DT	Decision Table
FIFO	First In First Out
FN	False Negative
FP	False Positive
GSN	Global Sensor Network
HVAC	Heating, Ventilation, and Air Conditioning
IAQ	Indoor Air Quality
IEQ	Indoor Environmental Quality
IRI	International Resource Identifier

JENA	A free and open source Java framework for building Semantic Web
KDSW	Knowledge Driven Sensor Web
KWh	Kilo-Watt hour
MAPE-K	Monitor, Analyze, Plan, Execute, Knowledge
MDP	Markov Decision Processes
MLP	Multilayer Perceptron
MQTT	Message Queuing Telemetry Transport
OGC	Open Geospatial Consortium
OS	Occupant satisfaction
OWL	Web Ontology Language
PM	Particulate Matter
POMDP	Partially Observable Markov Decision Processes
RDF	Resource Description Framework
RDFS	Resource Description Framework Schema
RF	Random Forests
RSP	RDF Stream Processing
SMS	Short Message Service
SPARQL	An RDF query language
C-SPARQL	Continous SPARQL: A version of SPARQL for querying data streams
SSN	Semantic Sensor Network
SSN-GX	Semantic Sensor Networks Incubator Group
SSW	Semantic Sensor Web
SWAP	Semantic Web Agent Platform
SWE	Sensor Web Enablement
SWRL	Semantic Web Rule Language
TN	True Negative

TP	True Positive
W3C	World Wide Web Consortium
WEKA	Waikato Environment for Knowledge Analysis
WHO	World Health Organization
WWW	World Wide Web
XML	eXtensible Markup Language
PM_{10}	Particular Matter of aerodynamic diameter of 10 microns
$PM_{2.5}$	Particular Matter of aerodynamic diameter of 2.5 microns

Chapter 1

Introduction

Observing and monitoring of the natural and built environments is important in many application scenarios for proactive control over environmental situations. The natural environment includes naturally occurring phenomena, features, and living and non-living things that affect human life and activity [52]. The built environment encompasses the structures and facilities formed by people for living and working [79, 140]. Environmental monitoring applications typically incorporate some sensor technology to continually observe specific features of interest in the physical environment and transmit data emanating from these sensors to a computing system for analysis. Sensor applications range from space-borne earth observation satellites to low-cost indoor sensors, including devices worn by people [32, 115, 167]. Environmental sensor applications can be impactful if the sensor observation data can be used to proactively take control over environmental situations in favor of the users.

An environmental monitoring application can be designed to be *reactive* or *proactive*. Reactive monitoring applications are designed to detect occurrence of specified situations and act in order to take control of such situations and minimize their effects in favor of the user. Proactive monitoring applications, however, can anticipate the future occurrence of the situation and execute control actions in order to avert the situation before it occurs. While reactive monitoring situations can detect situations that have occurred, from sensor data, proactive monitoring can anticipate future occurrence of situations, making it possible to prevent the situation from occurring. The goal of a proactive application includes analyzing sensor observations to detect situations of interest; to anticipate future occurrences of the situations; and to process and enact decisions [146].

1.1 Background

Data captured by sensors are known to be opaque with minimal contextual information, thereby necessitating semantic enrichment of the sensor data for analysis [135]. Thus, Semantic Sensor Web (SSW) technology is widely adopted for processing sensor data, as it provides tools and techniques for semantic enrichment and analysis of sensor data in order to make sense of it. In this thesis, the term *situation* is understood as an interpretation of sensor data in an application domain [164] and is used in the context of the state of monitored features in a physical environment. Hence, *situation analysis* encompasses the process of detecting (*situation detection*) and predicting (*situation prediction*) a situation of interest.

Semantic Web technologies, for example, ontologies, can be used to model concepts and relationships in a domain of interest [20, 134]. Standardization efforts in the SSW has led to the Semantic Sensor Network (SSN) ontology which has become the de-facto ontology for SSW applications [43]. Raw sensor observation data is annotated and encoded with semantic metadata, which allows for the integration and fusion of sensor data from heterogeneous sources. It also facilitates reasoning to make inferences about an observed feature of interest in the environment by evaluating semantic queries on semantically enriched data [34, 135]. SSW techniques have been investigated for monitoring and providing environmental decision support in different application domains [66, 105]. While some progress has been made in terms of classifying current situations of interest from streaming sensor data and taking responsive actions to mitigate these situations, predicting future situations for proactive control remains a challenge.

Some recent efforts have proposed semantic methods, such as predictive reasoning, in semantically annotated data streams [87, 92, 93] and, although it is an active research area, it is still quite young with new techniques emerging. Statistical machine learning provides advanced techniques which support applying algorithms to learn certain properties and patterns of data to predict future trends. This study suggests that incorporating *machine learning algorithms* in an SSW monitoring system will allow for determining proactive control actions to enhance or avoid specific future situations in many environmental monitoring application areas.

Proactive control in many computing applications has been implemented in an ad hoc manner for decades. There have been some recent efforts towards proactive architectures, especially in event driven architectures [57] and self-adaptive systems [9], but such systems lack the expressive semantics that SSW offers. In environmental monitoring applications expressive analytical techniques are necessary for enrichment, analysis, interoperability and dynamic integration of data streams from heterogeneous sources. Statistical machine learning can be incorporated in an SSW application for monitoring and control. Advanced planning and decision making mechanisms, such as Markov Decision Processes (MDP), can provide a robust mechanism for modeling a coherent and consistent decision process, which can be incorporated in SSW applications.

1.2 Problem statement

SSW applications have great potential to proactively forestall many environmental situations in both the natural and built environment [66]. Proactive monitoring and control of the environment requires combining different technologies to support situation detection, situation prediction and decision processing [27]. Although SSW technologies provide promising tools and expressive analytical methods for sensor applications [35, 135], a gap in current SSW frameworks is the lack of essential mechanisms required for proactive control, namely mechanisms for anticipating the future and coherent mechanisms for consistent decision processing and planning [27].

Also, application frameworks that provide tools and techniques for rapid application development is still a gap in the SSW community [44]. This research is focused on filling these gaps.

1.3 Aim and objectives

The aim of this research is to investigate how to incorporate statistical machine learning for situation prediction and MDP theory in an SSW framework for proactive environmental monitoring and control.

1.3.1 Specific objectives

The specific objectives of this research are

(i) To develop an ontology-driven framework for monitoring and control applications, based on the SSW principles.

(ii) To design an approach to incorporate statistical machine learning for situation prediction in the SSW framework for proactive monitoring and control applications.

(iii) To design an approach to incorporate MDP theory in the SSW framework.

(iv) To evaluate the proactive SSW framework using real-life use case scenarios.

1.4 Contributions of the study

A proactive monitoring and control application shows great promise towards advancing and maintaining lives by forestalling many environmental situations which afflict humans. This work proposes a framework that combines the advanced predictive capabilities of statistical machine learning with the expressive semantic analytical capabilities of SSW for proactive monitoring and control applications [5]. The framework also incorporates an MDP theory within the context of the SSW.

The contributions of this research are as follows: first, a framework that incorporates the required functional components to achieve proactive control in an SSW framework, the framework shows the design and implementation tools and techniques for a proactive SSW application [5]. Second, an approach to incorporate statistical machine learning models in an SSW for proactive environmental monitoring and control. The approach shows how to design machine learning for situation prediction and incorporate it into a proactive SSW framework. Third, an approach to enhance ontology-driven decision processing with an MDP to improve coherence and consistency in a proactive decision making application. And finally, a proactive monitoring and control ontology [4]. The ontology provides concepts to support the components of a proactive SSW framework. The ontology can be combined with a domain ontology in an application domain to develop a proactive monitoring and control application.

1.5 Overview of research design

Two different application use cases were chosen to evaluate the design and development of a SSW framework. This is to allow concrete real-world environmental situations to inform the design and evaluation of the proposed approaches and the framework.

1.5.1 Methods for achieving the research objectives

Proactive monitoring and control in the SSW is a challenging task which involves combining analytical methods and techniques. The first step was to review the state of the art in SSW, the proactive computing paradigm and the disciplines that support situation detection, situation prediction and decision processes in the extant literature. This is in order to elucidate requirements necessary for a proactive SSW framework. The requirements are distilled into areas of architectural concern and cross-cutting issues which drive the development of a proactive SSW framework. A basic prototype of the proposed model is first developed as an ontology-driven testbed, on which the approaches to incorporate other required components are to be evaluated.

The framework was evaluated on two different environmental use cases that are considered viable, with real-life sensor data.

Use case 1. The first use case is in the area of Indoor Air Quality (IAQ) for occupational health. IAQ is a growing health concern as exposure to indoor pollutants has been increasingly incriminated in causing illnesses, some of which are fatal [119, 130, 166]. The focus of this use case is proactive monitoring and control of occupants' exposure to indoor pollutants. This use case is carried out with an occupational health research group who are investigating the effect of indoor pollutants on pregnant women and children in an ongoing cohort study in South Durban, a low-cost residential area in South Africa.

Use case 2. The second use case is based in the area of demand side management (DSM) on the smart grid. Power demand management is a known problem that is currently receiving a lot of research attention [22, 143]. Load shedding is a known problem in many countries of the world where power generation could not meet consumption needs. This is especially true during peak demand periods due to extreme weather conditions [22, 143]. The focus of the use case is proactive monitoring and control of electricity use in the residential home environment to avoid power cuts and improve occupants' satisfaction.

1.5.2 Expected impact

Proactive SSW applications shows great promise towards improving automation in sensor-based monitoring applications. These are important in forestalling unwanted environmental situations before their occurrence. The potential benefits of this are as follows:

- Advocate a paradigm shift in the design and development of SSW applications from reactive to proactive ones. This is to provide seamless control of environmental situations before they occur.
- Promote proactive decision support in environmental monitoring applications.
- Provide a set of methods and tools to ease the development of proactive SSW applications.

1.6 Scope

This research is focused on approaches to incorporate statistical machine learning for situation prediction and MDP for decision processing in a proactive SSW framework. The target framework is situated mainly within the context of SSW, and the application use cases are mainly focused on monitoring and control in the physical environment. Hence, the framework is evaluated using two environmental use cases.

1.7 Thesis layout

The remaining part of this thesis is structured as follows:

- **Chapter 2 - Literature Review:** This chapter reviews the literature in the relevant fields that serves as background to this research, and recent approaches that have been proposed in related works.
- **Chapter 3 - An Abstract Architecture for a Proactive SSW Framework:** This chapter presents the design of an abstract architecture of the framework which resulted from the requirements that are extracted from the literature.

- **Chapter 4 - Ontology Driven Situation Analysis:** This chapter discusses the development and evaluation of a prototype of the proposed framework, an ontology-driven testbed on which the approaches to incorporate other components are to be evaluated.

- **Chapter 5 - Incorporating Statistical Machine Learning in an SSW Framework for Proactive Monitoring and Control:** This chapter introduces the proposed approach to incorporate statistical machine learning model in an SSW for proactive monitoring and control. The Chapter also presents the evaluation of the proposed framework with Use case 1 (see Section 1.5). The use case evaluates the framework with a focus on how to incorporate statistical machine learning in an SSW framework for proactive monitoring.

- **Chapter 6 - Incorporating MDP Theory into a Proactive SSW Framework:** This chapter presents the second use case for this research. The use case evaluates the proposed framework with the proactive monitoring and control of electricity use in residential blocks of houses. The chapter further introduces a proposed approach to incorporate rigor of a probabilistic decision-making MDP in the proposed proactive SSW framework.

- **Chapter 7 - Discussion and Conclusion:** This chapter discusses the results of this research. The evaluations of the proposed framework that are done in Chapter 5 and Chapter 6 are summarized, analyzed and compared to existing frameworks. The chapter further discusses how the objectives of this research are achieved, the feats of the proposed framework and highlights future directions.

‘

Chapter 2

Literature Review

This chapter reviews related literature that serves as background to this work. Since this research deals with combining different analytical techniques to manage streaming sensor data based on the proactive computing paradigm, this review covers a diverse range of disciplines, and identifies challenges that underpin the motivation for this work.

The Semantic Web is reviewed in Section 2.1 and Section 2.2 reviews the notion of the Sensor Web, while the SSW is reviewed in Section 2.3. Section 2.4 reviews the proactive computing paradigm and related paradigms. In Section 2.5 works relating to the different tools and techniques that are essential to support proactive control in a computing system are reviewed. Section 2.6 summarizes the chapter and highlights the gap that motivates this research.

2.1 Semantic Web

The idea of a Semantic Web was originally introduced as an extension of the current web [20]. The Semantic Web vision relies on using existing technologies such as extended markup language (XML) and resource description framework (RDF) to extend the World Wide Web (WWW) to a web of data, that is, shift emphasis from documents in the current web to data which is accessible and understandable by humans and machines. These tools support modeling a sharable conceptualization of a specific domain of interest, an *ontology*, on the Semantic Web [37, 134].

2.1.1 Ontology

An ontology can be used to model terms, that is, concepts and relationships between them in a domain of interest, and for structuring web resources [37, 145]. Several definitions of an ontology are presented in the literature. A widely adopted definition of an ontology in computer science is that of Gruber [67]:

"An ontology is an explicit specification of a conceptualization" [67].

However, Borst later redefined ontology as:

"... a specific, formal representation of a shared conceptualization of a domain" [25]

The latter focuses on the application of ontology in computational systems, where *"specific"* refers to clearly specifying concepts, relations, instances and axioms in the domain of discourse; *"formal"* refers to being machine readable; *"shared"* implies that a community has consented to the content of the ontology; and *"conceptualization"* implies that it is an abstract model of a domain [25]. They also characterized ontologies in several dimensions such as granularity, formality, generality and computational capability (see Table 2.1).

2.1.2 Semantic Web languages

A detailed review of early approaches to knowledge representation languages is presented in Pulido et al. [121]. The extended markup language (XML) was notably found useful to support separation of the web content markup from that of web presentation, but its semantics capability is limited [121]. The goal of realizing a Semantic Web drove the creation of newer approaches. The standardized recommended

Table 2.1: Characterization of ontologies, adopted from Borst [25]

Dimension	**Types**
Granularity	• Coarse-grained • Fine-grained
Formality	• Highly informal • Semi-informal • Semi-formal • Rigorously formal
Generality	• Top-level (upper ontology), • Mid-level (utility ontology) • Task ontologies • Domain ontologies • Application ontologies
Computational capability	• Heavyweight • Lightweight

knowledge representation languages for the Semantic Web by the World Wide Web Consortium (W3C) are *Resource Description framework* (RDF) [88], *RDF Schema* (RDFS) and the *Web Ontology Language* (OWL) [99].

- **RDF and RDFS:** The RDF [88, 98] recently revised and updated to RDF1.1 [46], is a W3C recommended language that supports encoding, exchange and reuse of information about resources on the WWW. The framework adopts XML syntax (RDF/XML), which imposes structural constraints for unambiguous representation of resources. RDF identifies resources with their Internationalized Resource Identifier (IRI) references. An RDF expression is a triple notation, consisting of subject, predicate and object, where the predicate (also called property) expresses the relationship between the subject and the object. A triple can also be expressed as a graph (see Figure 2.1), in which subject and object are nodes and the predicate is a directed arc connecting the two [88].

  RDFS is a semantic extension of RDF that provides mechanisms for describing groups of related resources and the relationships between them, such as classes and properties, including the domain and range of the properties [31]. The need for a more expressive language than RDF and RDFS led to the Web Ontology

Figure 2.1: RDF triple as a graph, adopted from Klyne and Carroll [88].

Language (OWL) [31].

- **OWL:** The web ontology language was developed to fill the need for greater expressive power than RDF and RDFS could offer. OWL is a family of three language variants, usually referred to as species [99]. The species of OWL includes OWL Lite, OWL DL, and OWL Full, with increasing levels of expressivity. OWL provides extended vocabulary for describing properties and classes, such as relations between classes (e.g. disjointness), cardinality (e.g. "exactly one"), equality and richer typing of properties [99]. OWL relies on XML Schema which is limited in expressivity; the need for more expressivity fuels the need for revision of the language [65].

- **OWL 2:** OWL 2, a revision of OWL, adds new functionality to the language. Some of the new features are referred to as syntactic sugar (e.g. disjoint union of classes), others provide more expressiveness [65, 72]. These include keys; property chains; richer data types and data ranges; qualified cardinality restrictions; asymmetric, reflexive, and disjoint properties; and enhanced annotation capabilities [65, 72].

2.2 The Sensor Web

The idea of a Sensor Web was originally conceived by Delin and Jackson [47] as a macro instrument:

"... a system of intra-communicating, spatially distributed sensor pods that can be deployed to monitor and explore new environments" [47].

They designed sensor pods equipped with transducers to convert environmental parameters to electrical signals, a wireless communication module for intra-communication between pods, a computing module for local analysis of the measured signals, and power source. Their concept of a Sensor Web was focused on the collective interpretation of signals from the coordinated sensing pods, over space and time, with the goal of extracting knowledge from data collected. This notion of a Sensor Web can be summarized as a local network of sensors deployed for a particular goal. The sensor network is not necessarily accessible via the Internet, nor the sensor observation data necessarily exposed over the WWW.

The notion of a Sensor Web soon changed from a localized macro instrument to an Internet-wide network. Gibbons et al. [64] proposed "IrisNet" with a vision of an Internet-wide sensor network, where computing devices on the WWW have sensors attached. They proposed a scalable agent-based architecture with two types of agents: sensing agents for collecting and storing sensor data and organizing agents for querying the data. IrisNet is known to be a resource-intensive approach to deploying a Sensor Web [36, 64].

Shneidman et al. [136] noted the need for an infrastructure to connect many disparate sensor networks to the applications that desire data from them. To fill this gap, they proposed Hourglass [136]. Their infrastructure consists of network-connected dedicated machines that provide service registration, discovery, and routing of data streams from sensors to client applications. Hourglass also allows for in-network data processing services. While Hourglass was able to provide an approach to deliver sensor data to applications as a service on the Internet, there was a need for a high level of abstraction on an infrastructure to enable simple and rapid deployment of sensor applications [2].

Aberer et al. [2] proposed Global Sensor Networks (GSN), a middleware that provides abstraction over the sensors and infrastructure details, with a goal to support simplicity, adaptability, scalability and rapid deployment of sensor applications. GSN was focused on sharing and integration of data among heterogeneous sensor networks on the Internet and minimizing deployment efforts. A notable abstraction in

GSN is the virtual sensor, which can be any form of data producer that takes in any number of input data streams and produces one defined data stream [2]. As several techniques evolved to support exposing sensor data on the web, interoperability between different infrastructures and the data model became a challenge due to lack of standards.

The Open Geospatial Consortium (OGC), a not for profit standard body of over 500 organizations from industry, academia and government, founded the Sensor Web Enablement (SWE) initiative to build frameworks of open standards for exploiting web-connected sensors and sensor systems [26]. SWE is a set of standard encodings and web services to support: discovery of sensors, processes, and observations; tasking of sensors or models; access to observations and observation streams; publish-subscribe capabilities for alerts; sensor system and process descriptions [26]. The SWE supports adding sensors and sensor data to the Internet. The SWE object oriented description of sensors and sensor data provides efficient generation of standard schema metadata for the observation data produced by sensors, and also facilitates the discovery and interpretation of archived data [117]. The SWE initiative was noted to lack semantically rich discovery mechanisms in the proposed services [137]. While the initiative solves the issues of interface heterogeneity, the issues of data and concept incompatibilities remained [42]. Hence the search for semantically enriched tools and techniques for analyzing sensor data [104, 135].

2.3 The "Semantic Sensor Web"

Sheth et al. [135] first coined the term Semantic Sensor Web. Their goal was to support the integration and communication between sensor networks, and thereby support knowledge acquisition from the integrated sensor data streams. They noted that sensor data, is by nature opaque and thereby propose semantic enrichment of the sensor data, to aid knowledge acquisition. Sheth et al. [135] noted that sensor encodings of observed phenomena as provided by the SWE initiative, which is usually presented in binary or proprietary formats, are by nature opaque and give minimal information. Hence, they proposed semantic enrichment of sensor data with spatial, temporal and thematic metadata, to facilitate expressive analysis. For example, sensor observations about phenomena such as temperature and precipitation can be annotated with time, location and theme, which can also be encoded as RDF triples (see Figure 2.2)[135]. Evaluation of complex semantic queries to derive higher-level information from the enriched sensor data is thus allowed. This approach leverages ontologies to describe concepts and the relationships between them, both in the sensor and the relevant application domain.

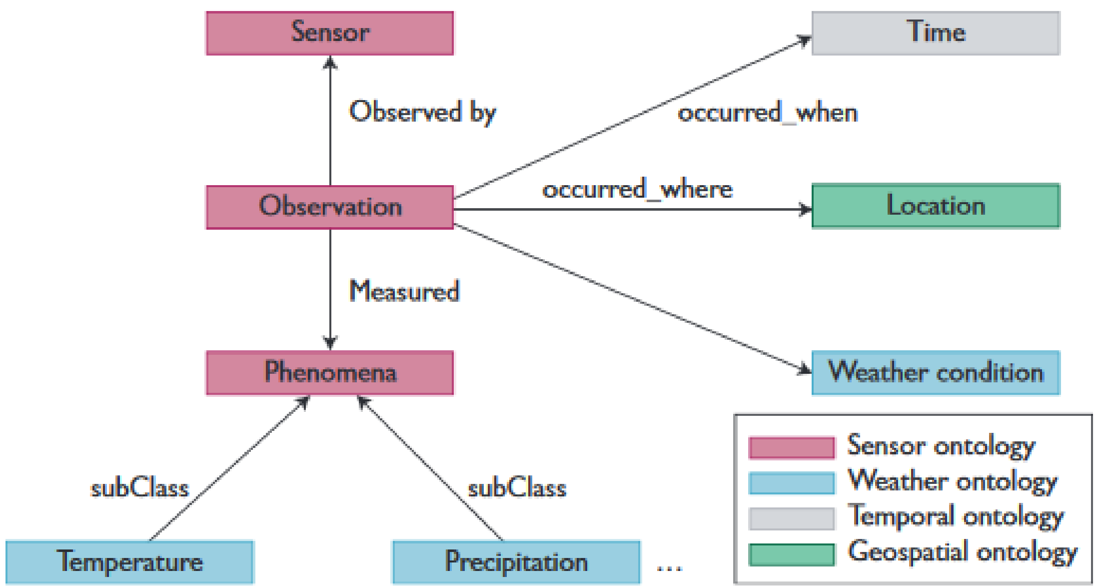


Figure 2.2: Annotating sensor data with metadata, adopted from Sheth et al. [135].

Approaches have been proposed to developing ontologies for sensor applications, focusing on different aspects of sensor systems [11, 55, 80, 129]. Some earlier approaches to model sensors with ontologies were not focused on managing sensor data; for example, Avancha et al. [11] proposed an ontology to manage sensor nodes. A survey of ontologies and semantic specifications that have been proposed for sensors and sensor networks is provided by Compton et al. [42].

2.3.1 The Semantic Sensor Network ontology

Drawing from the success of the previous approaches, the Semantic Sensor Network (SSN) Incubator Group of the W3C (SSN-XG) , which comprises members from the Semantic Web and Sensor Web communities, developed the SSN ontology. The SSN ontology is a generic description of sensor assets that can be applied to different application domains. SSN ontology is modular (see Figure 2.3), and it is compatible with OGC SWE services. The ontology features four main perspectives [43]:

- **Sensor:** The sensor perspective focuses on the sensor that senses, how it senses, and what is sensed;
- **Observation:** The observation perspective focuses on the observation data and related metadata. Observations are contexts for interpreting the stimuli;
- **System:** System perspective focuses on sensor systems as part of sensing infrastructure and their deployments.
- **Feature and property:** This is a focus on the particular property that is sensed and the observations that have been made about it.

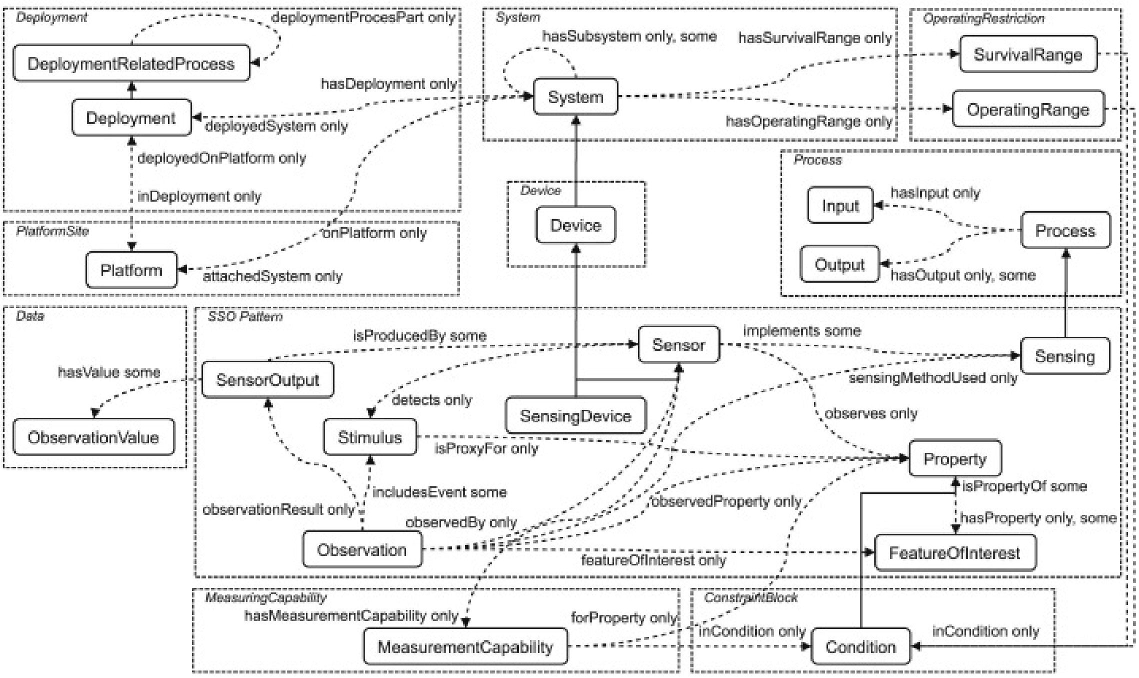


Figure 2.3: The SSN ontology, adopted from Compton et al. [43]

2.3.2 Ontology access to sensor data streams

Sensor data model: Sensor observation data is typically time-series data. Sensor data consists of measurements recorded over a time interval, generally referred to as data points recorded and as a sequence ordered by the time stamp of each record [34]. In real-time applications sensor data is generated continuously, and the data stream becomes unbounded. A large body of work exists in the areas of Database Management Systems (DBMS) [23, 147], and Data Stream Management Systems (DSMS) [12, 13, 49, 90], on the traditional approaches and methods that have been proposed for managing data streams. DBMS approaches provide a static method for managing data streams in which the dataset is expected to be persistent. The downside of the static approach is that when the data stream is generated at a high frequency, it becomes resource-intensive and may be unnecessary to store the unbounded data. DSMS methods support the use of continuous queries [13, 147] to manage continuous data streams. A query is issued once and the data stream filters through it continually.

Context information about the sensor observations, usually referred to as meta-data, helps to make sense of the recorded data. The early approaches in the Sensor Web adopted XML to model sensors and sensor data [2, 113]. XML provides a flexible method to represent the sensor data and corresponding metadata; it also supports interoperability of data from heterogeneous sources. The OGC SWE initiative standardization efforts and services were also based on XML data model for sensor data. However, XML based approaches are known to be limited in semantics; for instance, XML does not support explicit description of relationships between data resources [17].

The SSW approach introduced the use of RDF for modeling sensor data [103, 135]. Extending the XML-based data model to RDF triples and ontology primitives does not only bring more structure to sensor data, it also provides a semantically rich method for storing meta-data along with sensor observation data and explicit description of relationships between data components and attributes. It further allows for advanced analysis on the data along with concepts and relationships in ontology, for more expressive information recovery and reasoning on the ontology to infer new information [17]. This approach supports analyzing semantic queries such as SPARQL query on the stored data, for analysis and information retrieval. This approach to SSW is similar to DBMS where data is persistent in the database and the query is transient. Here data is persistent in an ontology or an RDF triple store. Streaming data is transient. In a real-time application, storing data before querying it can presents unbearably high latency, and when data stream is generated at a high frequency it may not be scalable to store all the data [5, 35, 48].

Stream reasoning: Recent research efforts in the SSW have been focused on approaches that provide ontology-based access to sensor observation data streams [16, 35, 48]. Stream reasoning approaches also referred to as RDF Stream Processing (RSP), support semantic methods to process streaming RDF data with continuous queries, in a manner similar to DSMS (see Section 2.3.2). Most of the continuous query languages present extensions to SPARQL query language, such as SPARQL-Stream [35]and C-SPARQL [16]. In the simplest form, the continuous data stream is divided into time windows, on which SPARQL queries are executed. Stream reasoning provides expressive methods to analyze streaming sensor data. It supports integration of data streams from heterogeneous sources. It also supports querying data streams with background knowledge in the ontology without necessarily having to store the data.

2.3.3 Architectures and frameworks

Gray et al.

Gray et al. [66] reported on an SSW for environmental decision support. The service-oriented architecture features the orchestration of ontologies and services focused on publishing, discovering and integrating sensor data for emergency decision support. Their architecture is related to the studies presented in this thesis in two main ways. First, their architecture was based on SSW principles, which advocates the use of Semantic Web technologies such as ontologies to process sensor data. Second, their work was focused on the physical environment that is providing early warning based on environmental monitoring. However, their architecture is different from this research in the following ways. Although their architecture attempts to provide early warning signs, based on data from discovered services which are external to the system, it does not emphasize the mechanisms for anticipating the future as part of the architecture [66]. Their work does not provide an approach to design and integrate situation prediction in the SSW architecture. Also no approach was provided for designing, incorporating and executing proactive decisions. Finally, their architecture does not follow the principles of proactive computing.

Choi and Rhee

Choi and Rhee [40] investigated a distributed SSW architecture which consists of two main components: an SSW Platform and a smart gateway. The architecture was focused on separating services domain and information processing. The smart gateway aggregates sensor data from which it generates context information for the SSW platform. The SSW platform discovers and aggregates services which use the context information to provide user-defined services, all within the principles of SSW. Their system is related to this research being an SSW architecture; however, it is a reactive SSW architecture, which reacts only to situations that have already occurred and been detected by the smart gateway. The architecture does not provide mechanisms to anticipate the future. Hence, the architecture does not offer required functionalities to avert occurrence of a situation ahead of time.

Moodley and Tapamo

Moodley and Tapamo [104] proposed a knowledge driven Sensor Web (KDSW), an ontology-driven multi-agent approach for designing Sensor Web systems. The proposed semantic infrastructure which is based on a Semantic Web Agent Platform

(SWAP) [103] is robust and capable of an Internet-wide Sensor Web system. This approach to Sensor Web caters for uncertainties, system dynamisms and provides support for representing agents, services and tasks. SWAP has been used in earth observation use cases [105]. Although the KDSW approach and SWAP have great potential for SSW applications, they are different from this work in the following way. The architecture has not been focused towards proactive control. Mechanisms for anticipating the future have not been integrated in the architecture. Hence it is only a reactive architecture.

2.3.4 Section summary

SSW provides rich sets of promising tools and techniques for developing sensor-based applications and for managing and processing sensor data streams. SSW has evolved from the union of activities from two research communities, namely the Sensor Web and the Semantic Web activity of the W3C. The earlier approaches in SSW are based on capturing data in an ontology or a data store and then querying them over temporal constraints, which creates processing overheads, while stream reasoning-based applications employ continuous semantic queries for processing semantically enriched streaming data on the fly. However, most of the SSW architectures are either reactive architecture which do not include mechanisms for predicting the future or rely on external predictive services. Furthermore, application frameworks to support rapid development of proactive SSW applications is still a gap [44].

2.4 Proactive computing

Sensor-based monitoring applications are increasingly deployed to support environmental control decisions and appropriate responsive actions about situations in the physical world. The proactive computing paradigm provides design principles and potentials that support monitoring and shaping of the physical world [146, 155]. The paradigm aims at bridging the gap between the virtual and the physical world by making computing artifacts equipped with sensors to understand the environment, anticipate the user's goal and act on his or her behalf [146].

The term *proactive computing* was introduced by Tennenhouse [146]. The paradigm aims to influence computing system design from the current interactive computing which was based on a human-centered principle, to a proactive one, where humans play more of a supervisory role rather than being the operator. The paradigm emphasizes reducing human interactions in the loop of automated process control. The three research loci for computing research to realize the vision of proactive computing are:

- **"Getting physical":** Connecting the physical environment to a computing machine is a primary requirement for taking control over it. Proactive computing advocates connecting computing systems to the physical world using sensors and actuators to both monitor and control the dynamics of the environment, in favor of the user [146].

- **"Getting real":** A proactive system needs to respond to external stimuli at a *"faster-than-human-speed"* [146]. The dynamics of the physical environment change at a speed higher than that which humans can sustain. Proactive computing advocates bridging the gap between control theory and computing systems. For example, computing operations such as software-enabled control, network-enabled control, online measurement and tuning, and the latency between input and output should be possible and in near real-time.

- **"Getting out":** While interactive computing has placed humans as operator in the loop of automated process control, the proactive computing paradigm promotes research for computers to switch into proactive modes of operations in which humans play more of supervisory role, rather than being the direct operator. This is essential to enable the system to respond to the physical world which changes at a speed humans cannot sustain [146].

2.4.1 Characteristics of a proactive system

Want et al. [155] characterized a proactive computer with three principles, based on the research loci defined by Tennenhouse [146]. These are: *connecting to the physical world*; *real-time and closed loop operation*; and *anticipation*:

- **Connecting to the physical world:** In order for computing systems to help in day-to-day activities, the physical world needs to be instrumented in such a way that they can have direct and intimate knowledge of the environment and use such knowledge to make changes to the real world.

- **Real-time and closed loop operation:** Integrating computing systems into the real world requires real-time operations. Having humans in the control loop of the computing process is inflexible. The system must be able to respond to events in the physical world much faster than is possible with a person.

- **Anticipation:** *"Anticipation is the cornerstone of proactive computing"* [155]. To be truly proactive, a computing system must predict the future in some sense. It must leverage context aware operations, statistical reasoning, and proactive data handling.

Proactive systems should be able to act in favor of the user (*pro-user*) and on their own (*autonomous*) [131]. Salovaara and Oulasvirta [131] further characterized proactive systems with five different attributes discussed below.

Real time operation: The system must follow an ongoing activity and its context in real-time.

World model: For a system to follow an activity and make inference, the system must have knowledge of the world model and the dynamics of its context. These may be explicitly represented or implicitly implemented in the system.

Hypothesized goal state: A system must know the hypothesis of the user's goal state in order to act on his or her behalf.

Sensitivities to future alternatives: The system must use its real-time information to make choices between different future alternatives.

Taking initiatives: The system takes actions on its own in favor of the user's goal. The vision of proactive computing is in contrast to the prevailing paradigms that have influenced the research and the design of computing systems, especially the *'man-computer symbiosis'* [95] paradigm which promoted the design of interactive systems. To clearly identify the goal of the proactive computing paradigm, in the following sections, the relationship between proactive computing and other computing paradigms is reviewed.

2.4.2 Related paradigms

In this section we discuss the relationship between the proactive computing paradigm and other paradigms that have influenced the design and development of computing systems.

- **Man-computer symbiosis**

  In the early years of computer development Licklider [95] introduced the paradigm of 'Man-computer symbiosis'. He envisioned a synergistic symbiotic interaction between humans and computers, where computers *"augment human intellect by freeing it from mundane tasks".* The man-computer interaction paradigm has largely influenced computing research in many disciplines, towards the design and development of personal computers, and office automation [14, 15, 59]. The paradigm essentially puts man in an interactive user mode with the computer; although this has been very successful in personal computing applications, as the processing speed of computers drastically increases, the limitations of the human's slow response speed become evident. For example, in the interactive mode the computing system has to halt several times to take in necessary data input or commands from the user for the next line of operation. Hence, the man-computer symbiosis paradigm cannot achieve the goal of integrating computer with the physical world. The dynamics of events in the physical world is complex. To integrate a computing system with the physical world, the system must be able to respond to real world events in real-time and at a much faster speed than possible with a person [146, 155].

- **Ubiquitous/Pervasive computing**

The notion of ubiquitous computing was introduced by Weiser [156], with a vision of computers 'disappearing' into many parts of everyday life, rather than just on the desk. The goal of ubiquitous computing is to integrate computer functionalities seamlessly into the physical world *"as a pervasive part of everyday life"* [156]. Ubiquitous computing has been referred to interchangeably as pervasive computing. A typical ubiquitous computing environment involves several microprocessor chips embedded into day-to-day objects in the environment, which are capable of sensing the environment, processing information and communications. The paradigm has facilitated research in many computing disciplines towards integrating computing devices and applications in different environments [3, 91, 115, 116, 139]. Proactive computing leverages and extends the ubiquitous computing paradigm in its aim to bridge the gap between the virtual and the physical world. However, integrating computing devices in the physical environment opened up other challenges in computing. This includes management of resources of the heterogeneous systems and communication networks [156].

- **Autonomic computing**

The autonomic computing paradigm which originated from IBM in 2001 was aimed at stimulating computing research towards computing systems that are self-managed [63, 73]. They envisioned emulating the autonomic nature of the human nervous system to realize computer systems that can manage themselves in the face of the rapidly increasing scale, complexity, heterogeneity and dynamism of networks, systems and applications [63, 73, 75, 85]. The main properties of self-management as described by this vision include: Self-Configuring, Self-Healing, Self-Optimizing, Self-Protecting [75]. To achieve the self-management of computing resources an architectural blueprint of an autonomic system and a reference model for an autonomic control loop was proposed. This is popularly referred to as the MAPE-K (Monitor, Analyse, Plan, Execute, Knowledge) [77]. A review of the autonomic computing research including the degrees of autonomy, autonomic models and autonomic applications, can be found in Huebscher and McCann [75].

The self-management perspectives of autonomic computing, and especially the MAPE-K control loop, are similar to the principles of proactive computing in two major ways with respect to their activeness. First, both paradigms support computing systems taking initiatives on their own to manage resources. Second, both paradigms support systems to pro-act, that is, to act on behalf of an entity. While the autonomic system principle is focused on acting to manage themselves, which makes the systems more dependable [142], a proactive system extends the autonomic system principle. More than managing its internal resources, a proactive system aims to interact with and shape the physical world [146, 155] in favor of the user.

2.4.3 Architectures and frameworks

Proactive control in computing systems has been implemented in an ad-hoc manners [56]. A review of some recent applications with such approaches is presented by VanSyckel and Becker [150]. However, there is a growing research interest in the design of proactive architectural frameworks for computing systems in different application domains [9, 27, 57, 118, 153, 166].

Engel et al.

One of the early approaches to model proactive control in a computing system was that of Engel et al. [57]. Although their work was based on event-driven computing, they provided a basic architecture for proactive computing systems. Their model, which was based on *"Detect, Predict, Decide and Act"* sequential orchestration of functionalities advocates the proactive computing principle. However, the architecture is different from ours as it did not employ SSW principles; neither is it an ontology-driven architecture. This research is focused on SSW technologies for environmental monitoring applications.

Anaya

Anaya [9] integrated predictive analysis in self-adaptive systems. The author proposed statistical machine learning techniques for predicting the future, and fuzzy logic for a control mechanism. Although the concept of our work is similar to Anaya's [9], who sought to achieve proactive control by integrating predictive analysis in a pervasive system, the author did not utilize semantic methods for analysis. Hence, the system lacks a mechanism to manage the model of the world. The architecture lacks the expressive analytical techniques of SSW technology which has been proved important [66] for environmental monitoring applications.

Yu and Lin

Yu and Lin [166] proposed an intelligent wireless sensing and control system to improve indoor air quality. Although their system design was focused on their application scenario, the orchestration of the system components is very similar to a typical proactive system. They used low-cost sensors for monitoring indoor carbon dioxide (CO_2) levels, they employed the integration of an Auto Regressive Moving Average (ARIMA) time-series forecasting method to predict future concentration levels of CO_2 from the sensor data and the system makes control decisions with a fuzzy logic component. The drawbacks of Yu and Lin [166] are as follows: first, the system does not provide a means to expressively model the domain, including the hypothesis of the user's goal, and the future alternatives, in a manner an ontology would do. All of these were implemented programmatically in the system, which will make reconfiguration and extension difficult if not impossible dynamically. Second, the work did not provide a generic architecture for designing proactive systems.

2.4.4 Section summary

The proactive computing paradigm was proposed to drive research towards the design and development of computing systems that could be connected to the physical world and pervade into day-to-day objects. Such systems rely on sensors and actuators to be able to monitor and shape the physical environment. The dynamics of the physical world is complex and faster than humans can respond to. Hence, proactive computing advocates reducing human involvement in the loop of automated process control, in contrast with the existing interactive computing where the human operator is a necessary part of the process. The paradigm extends the visions of related paradigms such as pervasive computing and autonomous computing. The immediate challenges of a proactive system are bringing together the technologies to support the functionalities required in order to achieve proactive control in computing systems.

2.5 Tools and techniques for proactive control

A proactive system is essentially a pervasive system equipped with a sensor that continually observes a particular property of a feature of interest in the physical environment [150]. The system also keeps the knowledge of the states of the world as interpreted from the sensor data, and the hypothesis of the user's goal's state. Achieving all these in a single framework requires integrating technologies for the functional components to analyze the sensor data [9, 27, 57]. First, the system requires technology to interpret sensor data to detect the state of the environment at a given time (*Situation detection*). Second is the ability to anticipate the possible future state (*Situation prediction*). Finally, based on its knowledge of the world, the system must be able to autonomously process decisions to select an appropriate action among alternatives in favor of the hypothesized user's goal state (*Decision processing*) [27, 57].

The remainder of this section discusses recent works and proposed approaches to achieve these functionalities in computing systems, with a focus on achieving proactive behavior in the systems.

2.5.1 Situation detection

Techniques for detecting situations from sensor data can fall into one of two major categories [164] as follows.

- **Learning techniques:** The machine is made to learn the complex relationship between the situation and sensor data and generate a model. This approach is known to require a lot of historical data [164].
- **Specification techniques:** These techniques require the modeler to specify the relationship between a situation of interest and data in a model to be encoded in the system. Hence, it requires that the modeler has a priori knowledge of the pattern of the situation to be represented.

The choice between learning techniques and specification techniques for situation detection is a trade off of requirements. That is lots of historical data and priori expert knowledge of the pattern of the situation of interest. Ontology-driven systems for situation detection, which is adopted in this research, is a specification approach that requires the domain expert knowledge of the situation to be represented in an ontology supporting the system.

2.5.2 Machine learning for situation prediction

Predicting the future occurrence of a situation of interest in order to enhance or reduce its probability of occurrence in favor of the user is the goal of a proactive system [146]. Predicting the future from sensor observations, time-series data requires time-series machine learning approaches. Sensor time-series data is sequential, hence, techniques that can handle time-series sequences are required for prediction. Two categories of approaches that have been proposed for predicting the future from sensor data include *predictive reasoning*, a knowledge representation method, and *statistical machine learning*.

- **Predictive reasoning**

  Semantic methods based on knowledge representation techniques are emerging to address predicting the future in sensor data streams [87, 92, 93]. Klarman and Meyer [87] employed temporal rules that accommodate complex data association patterns with logical and temporal constraints over DL-Lite Data streams for predictive reasoning. Lécué and Pan [92] addressed predictive reasoning from semantically annotated data streams by interpreting and correlating past semantics-augmented data over exogenous ontology and constructed predictions by cross-stream association rules. Although predictive reasoning is an active research area which provides expressive methods with promising results, it is young with new techniques still emerging.

- **Statistical machine learning**

  Statistical machine learning provides advanced techniques which support applying learning algorithms to learn certain properties and patterns of data to predict future trends. Sensor observation data is essentially time-series data.

  The classical statistical methods for modeling time-series data referred to as

Box-Jenkins Models were originally introduced by Box and Jenkins [28]. These methods involve identifying and specifying a model that fits the data. They have been used widely for predicting future values in sensor time-series data [166]. These include the Autoregressive Models, Moving Average Models, and the Autoregressive Moving Average Models, a mix of the first two, which are stationary models. The Autoregressive Integrated Moving Average is a non-stationary form of Box-Jenkins model [28, 29]. However, they have known limitations. Firstly, they assume that time-series data conforms to a linear model and follows a particular known statistical distribution, such as the normal distribution. Secondly, they assume that time-series data are either stationary or can be transformed to being stationary [86, 168]. Consequently, they are known to perform poorly when the continuity in the values of time-series data is uncertain [149].

On the other hand, a body of works exists on approaches that apply *data mining algorithms* for analyzing time-series data and do not require specifying a model for the data but rather generate the model from the historical data used in the training process [96, 102, 151, 162, 168]. These approaches have been grouped into three large categories known as feature-based, distance-based and model-based [162].

- **Feature-based approaches:** These approaches rely on properties that are representative of particular series to classify them. They transform a series of data into a feature vector onto which conventional classification methods are then applied to classify the series. The selection of the appropriate feature set is an important task in this approach. A simple form of this approach is to find a representative portion of the series that can be associated with a certain class [133, 165]. Other techniques include the use of kernel functions and heuristics to identify sub-sequences in the series that can be used to classify the series to a target class [41, 61, 70].

- **Distance-based approaches:** These approaches are based on using distance functions, a measure of the similarity between time-series data to classify them [1, 8]. Hence, choosing an appropriate distance function is the critical task in these approaches. Distance functions based on Euclidean distance have been found to be easier to compute and very useful in many scenarios. But their performance is known to be limited to the same length of series, and since they consider only one-to-one matches in the series, they are prone to errors even with a small misalignment in the series [84]. Other distance function approaches with more elastic matching include Dynamic Time Wrapping, which compares similarities of a point in a series with a small buffer of other points on the other series [50]. However, a drawback of this approach is that it is known to be computationally expensive. In general, distance-based approaches have been criticized for space and time computation limitations [161].

– **Model-based approaches:** Model-based approaches generalize classifiers into a certain model, based on the assumption that a class of time-series is generated by an underlying model [151, 162]. The task is to identify the model that can generate the sequences of the classes from certain parameters. Then the model learns the parameters from the training dataset and use them to assign new time-series to different classes. A large number of model-based approaches have been proposed for classifying time-series data. Examples of these include Bayesian Networks and Artificial Neural Networks [162].

A major drawback of using data mining approaches on streaming sensor data is that most of the approaches are originally designed for static time-series, that is, time-series in which all the data points are available at the time of processing [102, 149]. Streaming sensor data is unbounded and evolving. Furthermore, in a dynamic environment the relationship between the data and the properties of the target variable which the models predict is known to drift over time (*concept drift*) [62]. Hence, approaches that can learn continually and update the classifiers dynamically with the most recent data are required [51, 102].

- **Sliding window techniques for sensor data prediction**

In an effort to use data mining algorithms for prediction tasks on sensor data, some recent efforts have proposed the sliding window approach [102, 108, 149] for classification on time-series data. A *sliding window* is a fixed length of data that slides through the temporally ordered data stream [60, 149]. Sliding windows can be useful for two main purposes in time-series data classification tasks: first, to continually filter through the streaming data and select a fixed size of the most recent attributes as input for the classifier for predictions; second, to continually slide through historical data and select fixed size of data to update the classifier. Sliding window techniques are among the approaches that have been proposed to overcome concept drifts in dynamic time-series data [51]. Hence, the sliding window technique with data mining algorithms is adopted for situation prediction in this research.

2.5.3 Decision processing

Real-time decision processing in order to take control of anticipated situations *(proactive control)*, based on sensor data is a key challenge of a proactive application. Approaches that have been proposed for structuring decision processing in computer systems include ontology-driven approaches [39, 127, 154, 160], Bayesian decision theory [19, 27, 54], MDP [38, 124, 125], and Fuzzy set theory [53]. The latter, however, has no structure of its own and is rather used to augment the others to cater for vagueness in input data for decision making [53].

- **Ontology-driven approaches**

Increasing research efforts are exploring the use of Semantic Web technology, especially an ontology to model decision processing [39, 127, 128, 154, 160, 169].

Rospocher and Serafini [127] proposed using ontological representations of the data in a decision processing system to structure the dataset and to provide a content exchange format between the modules of the system; and to track intermediate data and results of the decision processing. They proposed a framework and an ontology for decision making which is comprised of three main components, namely, *problem*, *data*, and *conclusions* [128]. Using an ontology to structure decision processing in this manner has been noted to have several advantages [127, 128, 154]. First, both the knowledge and data for making decisions, which can exist in heterogeneous formats, can easily be combined for decision processing. Second, the decision processing which can be implemented as a service can also be combined with other semantic services that are available. Furthermore, this decision processing approach is known to support advanced reasoning techniques, which results in high-quality decisions.

A known drawback of ontology-driven systems is the lack of adequate support for uncertainties, which may impair coherence and consistency in decision outputs [10]. This inadequacy may be exacerbated, especially in the uncertain environment which characterizes sensor data. The classical decision theory is built upon axioms of probability and utility. Probability supports the framework for coherent assignment of beliefs with incomplete information, and utility theory provides a set of principles for consistency in processing preferences and decisions [71, 74].

- **Bayesian decision theory**

Bayesian decision theory provides a theoretical framework for modeling action and inference under uncertainty [18, 19, 27, 54]. The framework can be represented by an influence diagram, a directed acyclic graph, in which the decision variables are represented by nodes and the relationship between two nodes is represented by directed arcs connecting the nodes. Three different main types of nodes can be represented, such as chance node, decision node and utility node. A local probability distribution is maintained at each of the chance nodes and an utility table for the utility node. The decision node consists of the decision of alternatives. The alternative that maximizes expected utility, given the different probability distributions in the chance nodes, is considered the best decision. The Bayesian theory approach has been noted to be useful for decision making problems with several structured decision variables [19, 27].

- **Markov decision processes**

MDP and the Partially Observable Markov Decision Process (POMDP) have proven useful as rigorous mechanisms for modeling planning and decision making in uncertain environments [38, 107, 124, 125, 148]. MDP can be used for planning and decision making problems in dynamic environments. The decision processing task is to find an optimal *policy* which is a sequence of actions (given the state of the system) that maximizes the expected *reward* over a defined time *horizon* [148].

MDPs have been explored for making proactive decisions with promising results [57, 107]. However, the support for uncertainties in MDPs is known to be limited to fully observable domains [38, 148].

POMDP, which is a generalization of MDP, caters for even more uncertainties by the introduction of the notion of a *belief* of being in a state. A basic POMDP model consists of: a finite set of states; a finite set of actions; a finite set of observations; a state transition function, which is a probability distribution over some finite set of states; an observation function, that is, a probability distribution over some finite set of observations; and an immediate reward function [38, 148].

The complexity of computing optimal policies in MDPs and POMDPs is known to be very high due to the extensive computation involved in calculating the utility and beliefs, which suggests why they are often avoided in real-time systems. Some recent works have proposed algorithms and approaches to minimize the running time and make them suitable for real-time applications [107, 123, 148].

Some proposed approaches have attempted to leverage the combined advantages of more than one existing theory for decision making. In this manner, Bayesian decision theory has been combined with an ontology [10, 171]. However, there have not been many efforts in integrating MDP processes in this manner.

2.5.4 Section summary

Proactive control in a computing system depends on three main functionalities; situation detection, anticipation and decision processing. The system also requires a mechanism to manage the knowledge of the real world, the hypothesis of the user's goal state, and alternative actions to shape the world dynamics. Hence, an architecture that brings together all the technologies to support these functionalities in an SSW framework is required to achieve proactive monitoring and control within the context of SSW.

2.6 Summary

This chapter has presented a review of the existing literature in diverse paradigms that support technologies necessary to achieve the objectives of this thesis. The review started with the evolution and the state-of-art in SSW technologies. The related work in the proactive computing paradigm and previous techniques used to achieve proactive control in computing systems were also reviewed. Finally, to identify the gap which this thesis attempts to fill, existing related architectures were reviewed.

The design and development of proactive systems is currently gaining attention in different application domains. SSW technologies provide a rich set of promising tools and techniques for developing sensor-based applications, but most SSW applications are still designed in a reactive manner, and current SSW architectures are not based on proactive computing principles. Although there have been some recent efforts proposing proactive architectures, such models do not employ SSW technologies. They lack the expressive semantic analytic techniques that SSW offers. There is no current SSW architecture that offers all functionalities required to achieve proactive monitoring and control in a single framework.

This review reveals two main gaps which this research is designed to fill. First, current SSW frameworks lack essential mechanisms required for proactive control. These are mechanisms for anticipating the future, and coherent mechanisms for consistent decision processing and planning. Secondly, lack of frameworks for rapid SSW application development is still a known gap in the SSW community.

The next chapter discusses an abstract architecture for a proactive SSW framework based on the proactive computing paradigm.

Chapter 3

An Abstract Architecture for a Proactive SSW Framework

The main goal of this thesis is to investigate a SSW framework for proactive monitoring and control applications. To achieve this, we designed an abstract architecture to enhance and simplify the incorporation of situation detection, situation prediction and proactive decision mechanisms in the framework. Part of this chapter has been reported in Adeleke and Moodley [4].

As a point of departure, we considered the SSW framework [103, 135] and the proactive computing paradigm originally introduced by Tennenhouse [146], with their interpretations as reported in the literature (see Chapter 2). We focused on the basic characteristics of a proactive system [131], and functional components of a SSW framework [135]. From these (see Section 2.3, Section 2.4), we elicited four areas of concern necessary to achieve proactive control in a SSW monitoring and control application, namely, monitoring, situation analysis, control, knowledge management.

This chapter is organized as follows. In the next section we discuss the design goals and the areas of concern. Section 3 discusses the abstract architecture of the framework, while Section 4 presents the use of ontology to drive the framework.

Finally, Section 5 concludes with a summary of the chapter.

3.1 Design principles

Design principles are used in software architecture to guide the principal design decisions involved in the software design [100]. The design of the framework is governed by five core requirements, these are listed as follows.

(i) Adequate representation of all the areas of concern to achieve proactive control in environmental monitoring applications.

(ii) Proactive control represented in line with the proactive computing paradigm introduced by Tennenhouse [146].

(iii) Implementable within the context of the SSW.

(iv) Extendable and allowing reuse of relevant existing components, that is, allows incorporation of more functional components.

(v) Applicable in solving different environmental monitoring and control problems, that is, not application specific.

3.2 Areas of concern

The four AoCs considered to determine the required components of the proactive SSW are monitoring, situation analysis, control and knowledge management. The latter is treated as a cross-cutting concern. These are discussed as follows.

3.2.1 Monitoring

A proactive system is required to continually monitor a property of the feature of interest in an environment and analyze the recorded observation data in near real time. The SSW is essentially an ontology-driven application, which leverages ontologies to manage sensor observations in an application domain and enrich the sensor observation data with semantics to aid analysis. Hence, monitoring in the framework includes support for sensors, the sensor platforms, communication devices, the processing system components and the ontology modules that are involved in the capturing, recording, transmission and enriching of the sensor data. It also includes static data that are pre-captured in the ontology, such as expert knowledge and relevant metadata for semantic enrichment of the sensor data.

3.2.2 Situation analysis

A proactive system is essentially required to have a model of the world and a hypothesized user's goal state [131]. These requirements in turn require specifying some *indexes* that can facilitate the translation of quantitative sensor data values to certain qualitative states of the world, and thereafter, analyzing the semantically enriched streaming sensor data in order to detect the current state and to anticipate possible future occurrences of such situations. Situation analysis requires two major functionalities, namely, *Situation detection* and *Situation prediction*.

- **Situation detection:** In an ontology-driven framework, situation detection employs specification techniques in which the relationship between the target situation and sensor data is expressively encoded in the terms of the ontology (see Section 2.5.1). At run time, queries are evaluated over the ontology to combine the enriched current stream of sensor data with the assertions in the ontology in order to detect the situation which the current data represents. The traditional method is to store the enriched data stream in the ontology in order to combine both data and the rules in a query. However, recent efforts in stream reasoning allows ontology based access to streaming data in which the enriched streaming that can be combined with static data and assertions in the ontology on the fly (see Section 2.3.2).

- **Situation prediction:** Anticipation is a cornerstone of achieving proactive control in a computing system [146]. Situation prediction involves anticipating possible future states of the world over a specified horizon of time. Although some recent efforts are exploring predictive reasoning, which is based on knowledge representation techniques with promising results, in this work we employ Statistical Machine learning based situation prediction (see Section 2.5.2). A statistical machine learning technique can be used to model the states of the world and learn the complex relationship between the data and states of the world from historical streaming data. This can be used to predict the state of the world for the next time-step into some defined classes of possible states.

3.2.3 Control

The goal of a proactive system is to make decisions and act, based on an anticipated situations, in order to reduce or enhance the probability of occurrence of the situation, on behalf of its user's goal [146, 155]. That is, to control the system based on the hypothesis of the user's goal state [131]. Two important functionalities are required in this areas of concern, namely, decision processing and action.

- **Decision processing:** This is necessary to select an appropriate action among alternatives in order to control the system in favor of the hypothesis of the user's goal state. Ontology-driven decision processing involves evaluating queries on the ontology to activate appropriate rules and infer a corresponding decision based on some specified logic. The classical decision theory is based on the axioms of probability and utility. The concern here is to provide support for a coherent and consistent decision mechanism for the use case at hand.

- **Action:** Control actions are actions that have either direct or indirect influence on the dynamics of the states of the world. The actuating components of the system then give support to committing resources to the decision output.

3.2.4 Knowledge management

Knowledge management functionality is required across all the other areas of concerns in the framework. Hence, it is treated as a cross-cutting concern in the design of this framework. An ontology provides adequate proven techniques that can be used to model the knowledge requirements of all parts of the system such as: sensors and sensing platforms; the contextualized states of the world; the hypothesis of user's goals; future alternatives and control actions. The SSN ontology can provide adequate support for the sensor-related concerns, while a domain ontology can model the world with its contextualized states. Functional ontology modules can also be designed to support the various functional components in all the areas of concerns.

3.3 Abstract architecture

The requirements in the areas of concerns, as discussed in the previous section, are combined to develop an abstract architecture for the framework. The model consists of three layers abstracted from the areas of concerns, namely, monitoring, situation analysis and control, each of which represents the system components and the ontology modules that support functional requirements of the areas of concern. Figure 3.1 illustrates the abstract architecture for the framework.

The abstract architecture consists of three layers:

- **Monitoring:** This serves as the interface between the framework and the monitored environment where sensor observation data on the features of interest are captured. It represents certain parts of the system and ontology module that support data and measurements, including both the streaming sensor observation data and pre-captured static data in the system.

- **Situation Analysis:** It represents parts of the system and the ontology module that support situation detection and situation prediction, the two processes

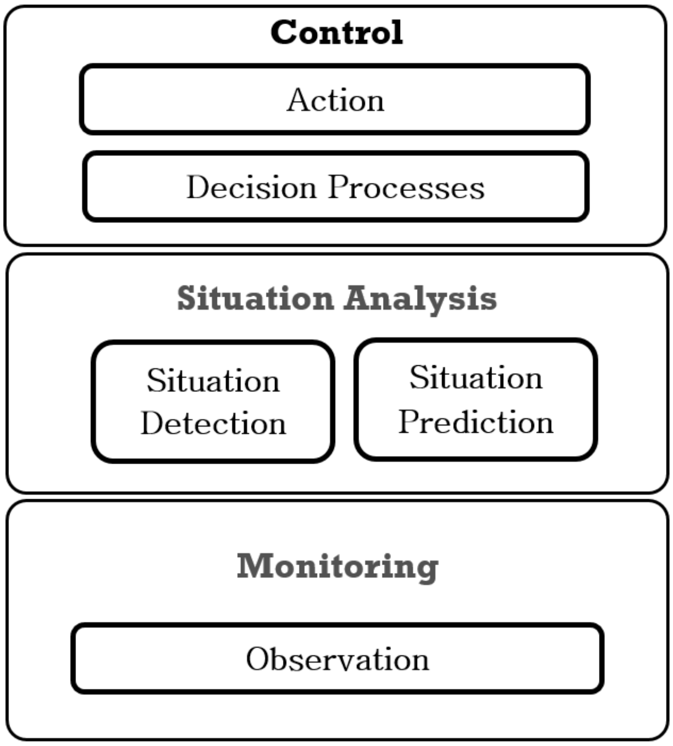


Figure 3.1: Abstract architecture for a proactive SSW framework.

that generate the current and future states, respectively. Situation analysis consists of two sub layers:

- ***Situation Detection:*** This sub-layer supports the detection of situations of interest in the system from the enriched sensor observation. A critical element of this component is the index that translates sensor data values to specified states.

- ***Situation Prediction:*** This sub-layer represents the part of the system which enables the prediction of the future states.

- **Control:** This layer consists of two sub-layers that use the predictions to create decisions and that transform the decisions into actions that can be carried out by either human or computer agents.

  - ***Decision Processes:*** This sub-layer represents parts of the system that are involved in deciding the control action to take, given the predicted future states. This layer fuses the identified current situation with the predicted situation to evaluate the most appropriate course of action.

– ***Action:*** This sub-layer represents parts of the system and ontology modules that supports encoding the control actions in the system, including support for enactment of the selected control action that corresponds to the result of decision process.

3.4 Data flow

In Figure 3.2 we illustrate the data flow through the main components of the system. Streaming data for proactive monitoring requires processing on the fly where the output of a process is automatically channeled as input for the next process.

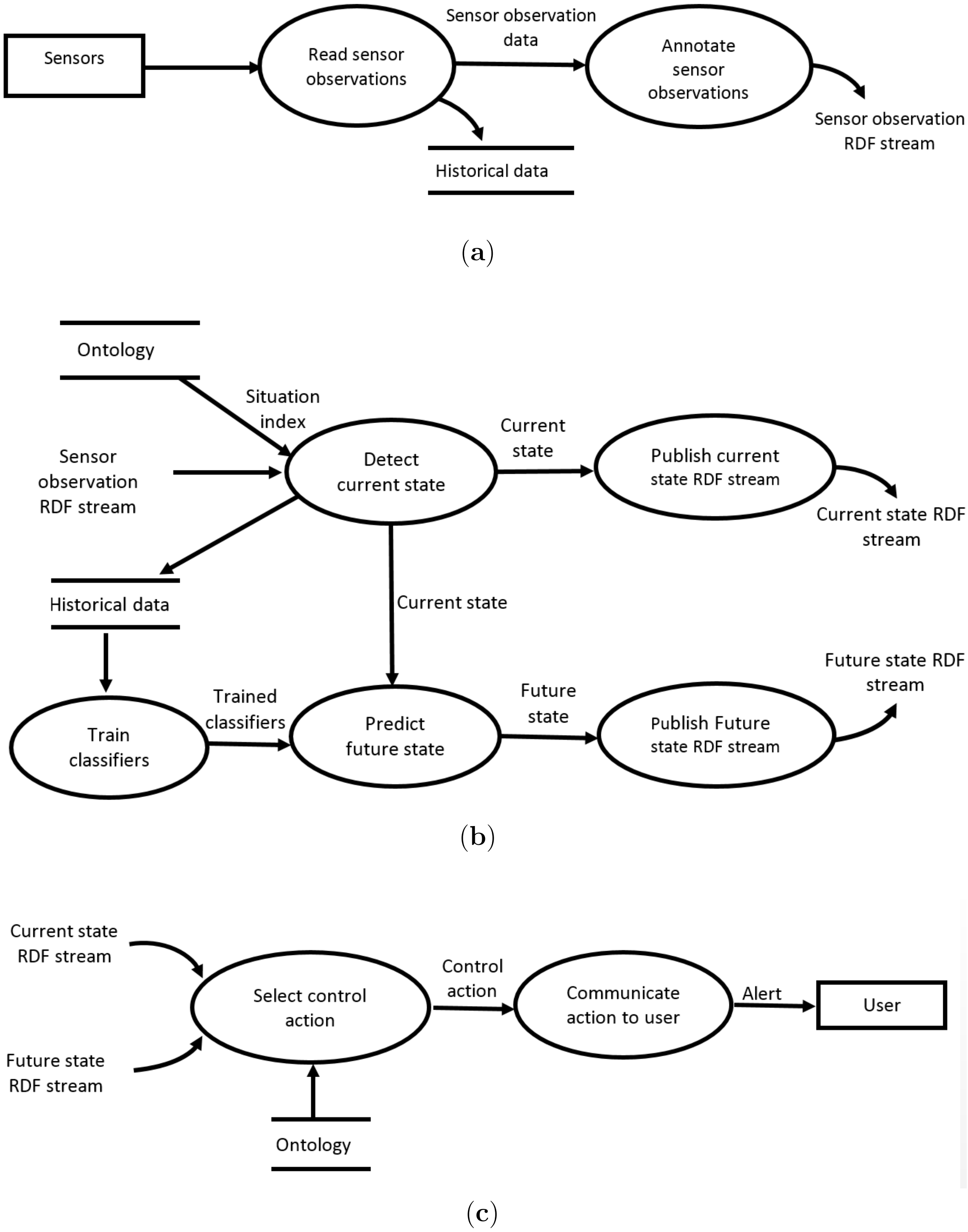


Figure 3.2: Level 1 Dataflow diagram of the proactive SSW framework: (**a**) Monitoring (**b**) Situation analysis (**c**) Control

However, historical data can be stored for later use, and temporary storage can be used to structure data as input for the next process. The observation data is streamed into situation detection and situation prediction components. The outputs of these two are integrated for decision processes. The decision output is then used by the action component to select the appropriate control actions to influence the dynamics of the state of the world.

3.5 Summary

In this chapter we have presented the design of an abstract architecture for a proactive SSW framework. The design started with the SSW framework [103, 135] and the proactive computing paradigm. From these, we identified areas of concerns needed to achieve proactive control in an SSW framework, which is governed by five core design principles. The areas of concerns are then distilled into a three-layered abstract architecture. Finally, we illustrated the data flow in the proposed framework with a data flow diagram.

The efficacy and use of the framework is evaluated through the development of two different prototype applications for environmental monitoring and control. These use cases are presented in Chapters 5 and 6. First however, to motivate the implementation strategies, in the next chapter, the development and evaluation of an ontology for monitoring and control is presented. This serves to provide support for incorporation and evaluation of other functional components of the framework.

Chapter 4

Ontology Driven Situation Analysis

This chapter presents an ontology-driven framework for environmental monitoring and control which was developed based on the abstract architecture presented in Chapter 3. The ontology-driven framework also serves as a testbed for the incorporation and evaluation of other required components for the proposed proactive monitoring and control framework. Parts of the study in this chapter have been published [4].

The development of the ontology is evaluated on an indoor air quality use case with real-life sensor data. Monitoring and control of Particular Matter of aerodynamic diameter of 10 microns (PM_{10}) in the indoor environment was chosen as the focus of the use case. Although the system for the evaluation represents the three layers of the proposed framework, at this stage, it does not include situation prediction. The development of the ontology is focused on two main sub-ontologies. The first is for proactive monitoring and control and the second is for indoor air quality. The ontology reuses terms from existing ontologies such as the SSN ontology [43] to support the monitoring layer of the architecture, CISRO time ontology [45] for temporal terms and the activity ontology [1] for activity related terms.

The rest of this chapter is organized as follows. In the next section we present the motivation for the study, followed by the use case in Section 3. Section 4 presents the development of the ontology, while Section 5 presents evaluation of the ontology.

The chapter concludes with a summary in Section 6.

4.1 Application use case

Detecting unhealthy indoor environmental situations and determining control actions to mitigate it requires a wide range of multidisciplinary domain knowledge. This includes sensor data processing, indoor air quality, and occupational health. Ontologies have been well investigated and found to be useful in capturing background and expert knowledge from different perspectives, integrating data from heterogeneous sources, building knowledge bases, knowledge acquisition (reasoning), analyzing data streams and managing knowledge and system dynamisms [35, 81, 104, 106, 114].

A crucial task in monitoring and control of an environmental situation is the detection of the particular situation of interest. This study addresses the situation detection with an ontology-driven Air Quality Index which can be generated by reasoning on the ontology. The ontology is evaluated by populating it with test data and querying it to analyze indoor environment situations relevant to the targeted use case scenario.

This use case is based on an ongoing cohort study in communities in the highly industrialized south Durban area in South Africa [78, 112] in which occupational health experts investigate the various effects of exposure to indoor air pollution on pregnant mothers and children. The area is generally inhabited by low income households. Occupants of houses in such setting are at high risk of indoor pollution. Houses in these areas have the following characteristics in common:

- Mechanical heating ventilation and air conditioning (HVAC) systems are usually not available, therefore ventilation is mainly by natural air infiltration through openings such as windows, doors and vents.

- Indoor activities that aggravate high levels of pollutant concentrations include smoking, burning fossil fuel for heating and cooking and burning of incense.

- Nearby light- and heavy-industry also produce harmful emissions that contaminate indoor air.

The goal of the occupational health care researcher is to maintain good ventilation, thermal comfort and harmful indoor air pollutants at acceptable levels.

4.1.1 Monitoring Particulate Matter

In this setting, occupational health researchers monitor the exposure of sensitive or vulnerable occupants of such buildings by placing expensive and cumbersome pollutant monitors (gravimetric particle monitors) in the indoor environment of the building for a day or two which are then removed and taken to the laboratory for analysis. This method is known as gravimetric analysis of Particulate Matter. [172]. During the procedure, occupants are also asked to complete activity diaries to collect indoor activity data for analysis. These are then used to understand patterns of air pollution.

4.1.2 Limitations of monitoring Particulate Matter

There are several limitations and challenges with the current process.

- It is cumbersome, capital intensive and dependent on manual processing. An occupational health officer needs to physically travel to each house to observe the equipment daily during the monitoring period.

- The analysis of the result takes several days, therefore it is inadequate to help occupants abate present situations as the result of analysis will only be available some days after the monitoring is completed.

- Because the monitoring is not continual, this approach assumes that the air quality of a monitored house is the same as measured during the monitoring time, which is actually not so in real life.

- The current process has low temporal resolution, as some of the pollutant monitors need several hours usually more than 8 hours to collect enough mass for analysis [163, 172].

The ontology-driven indoor air quality system attempts to deal with these limitations with a continual monitoring and control approach that alerts the occupants of unhealthy situations and suggests control actions to abate the situation in near real-time.

4.1.3 Key terms

(i) **Indoor environmental quality (IEQ):** A broad phrase that describes *"a building's environment in relation to the health and well being of those who occupy space within it"* [76]. In this work, the usage scenario is focused on two main aspects of IEQ. Indoor Air Quality and Thermal Comfort.

(ii) **Indoor Air Quality (IAQ):** A qualitative measure of the totality of the attributes of indoor air in relation to occupants health and well being [33].

(iii) **Air Quality Index (AQI):** A quantitative measure of the current level of pollution in the air based on the concentration levels of pollutants monitored [78]. An index is a range of values that corresponds to different scales of concentration level and possible health effects of each index. In this work we adopted the AQI by United States Environmental Protection Agency (EPA) [101].

(iv) **Thermal Comfort Index:** The level of satisfaction that people feel with respect to heat level of the environment.

4.2 An ontology for indoor air quality monitoring and control

In this section, we present the design of the ontology-driven system for proactive monitoring and control.

4.2.1 The system

The monitoring and control system is developed as a prototype of the SSW framework. In the use case scenario (see Section 4.1), low cost sensors were installed in the houses to observe pollutant levels and transmit observations to a central server for analysis in near real-time. The system was designed to analyze the data by querying the ontology. When an unhealthy situation is detected, it can raise an alarm and send feedback messages to the occupants if there is need to take action in order to abate the poor indoor air quality. Temperature, humidity and Particulate Matter (PM_{10}) sensors were installed on a Raspberry PI and Arduino prototyping platform, to capture relevant data. Figure 4.1 illustrates the proposed IAQ monitoring and control system.

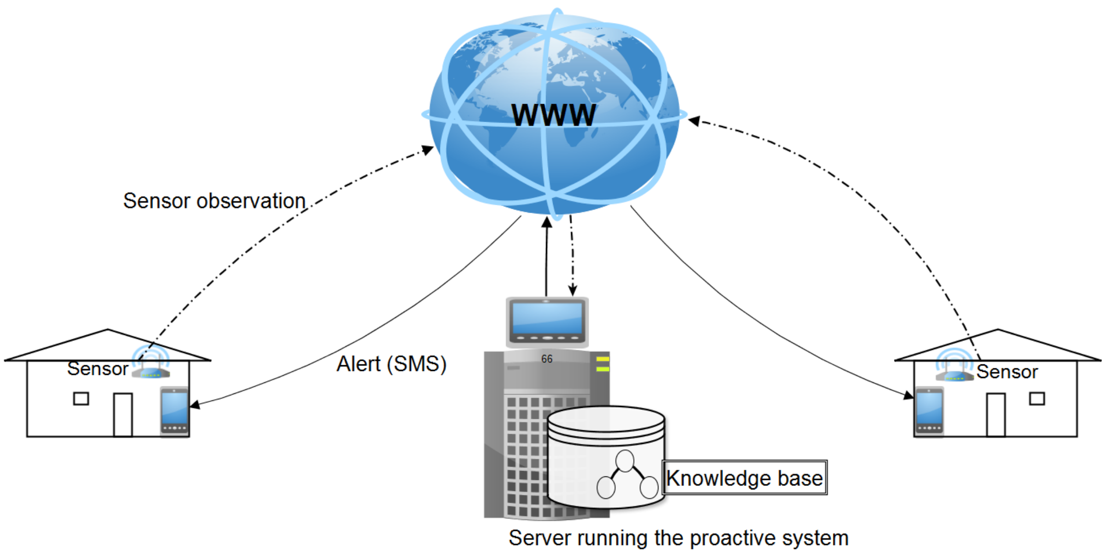


Figure 4.1: IAQ monitoring and control system

The system in this study was designed as a partial implementation of the abstract architecture discussed in Section 3.1. However, the situation analysis layer (see dotted lines in Figure 4.2) consists of situation detection components only. Hence, the system at this stage does not anticipate the future, but reacts to the situations detected in favor of the user's goal (*reactive*). Figure 4.2 shows the abstract architecture for our proposed framework and highlights the focus of this chapter, the dotted lines highlights the layers, the green line highlights the components focused on in the layer, while the grey box highlights the part that is not yet included in the framework.

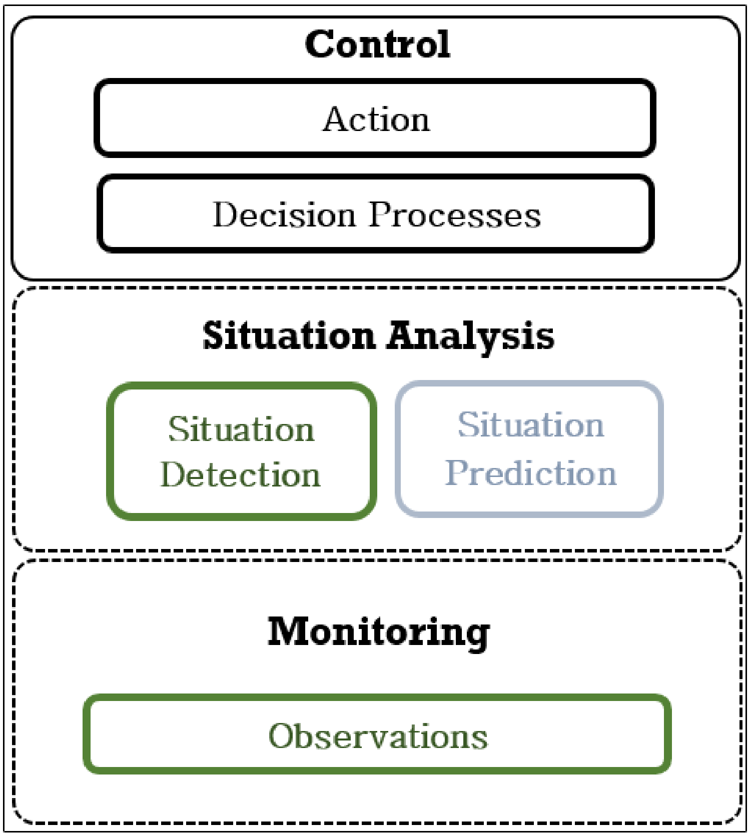


Figure 4.2: Abstract architecture for a proactive SSW framework.

The *monitoring* layer captures measurement and observation data. It provides representation support for building environment description, sensor observations and the occupant's activities in the indoor environment. While the environment description provides such data as features in the building, their type and quantity (for example number of windows, type of ceiling), sensor observations provide values of sensor measurements of the properties (temperature, humidity, pollutants concentrations) of the environment. The activity component provides a list of activities that impact positively or negatively on the air quality or thermal comfort of the environment.

The *situation analysis* layer provides representation and reasoning support for situation detection and classification. The Air Quality Index and Thermal Comfort Index which are sub-layers of this layer provide levels of abstraction over observation and measurements. The indexes allow for classifying quantitative observation values to qualitative states (*situations*) in the environment.

The *control* layer describes terms to determine whether any control action is necessary to abate a possible harmful situation. Although our approach is not geared towards activity recognition, capturing known activities relevant to the indoor environments in the target communities provides crucial knowledge for control feedback that can be exploited to abate poor indoor environmental situation. Some activities are known to induce pollutants, for example smoking indoors and cooking with paraffin stoves are known to increase indoor particulate matter concentration. This approach is especially useful in low resource settings where complex sensors and activities recognition systems are not available.

4.2.2 Design goal

The development of the ontology revolved around three core requirements in line with the abstract architecture. These are stated as follows.

(i) Adequate representation of sensor measurements pertaining to intended use case.

(ii) Representation and reasoning services for analyzing sensor data for detecting target situations.

(iii) Determining possible control actions to mitigate detected unhealthy situations.

4.2.3 Competency questions

Competency questions (CQs) are requirements presented as questions which an ontology must answer [68]. The ontology must contain sufficient terms and axioms to satisfy the CQs. CQs are useful for two main purposes. First, to enable developers identify important terms to create the ontology. Second, CQs provide a means to verify that the requirements of the ontology are satisfied [21].

The competence questions for the development of the environmental monitoring and control ontology are itemized as follows.

CQ1 Can the ontology adequately represent sensor data pertaining to the use case?

- Is it possible to stream sensor data to the ontology in near real-time?
- Can the ontology enrich the data with necessary meta-data for analysis?

CQ2 Can the ontology support adequate reasoning services for analysis of the sensor data to detecting targeted situations?

- Can the ontology automatically classify the quantitative sensor data to qualitative states to identify targeted situations?
- Is it possible to query the ontology for the classified states in near real-time?

CQ3 Can the ontology determine possible control actions to mitigate unhealthy situations?

- Can the ontology represent expert knowledge for determining control actions?
- Can the ontology match detected situations with appropriate control actions?
- Is it possible to query the ontology for the control actions in near real-

time?

4.2.4 Ontology engineering

Methonthology [58], the popular ontology engineering method was adopted to develop the ontology because it provides sufficient details, supports prototyping, and allows reuse of existing ontologies. Methontology proposes seven activity phases for use in building an ontology, namely, specification, knowledge acquisition, conceptualization, integration, implementation, evaluation and documentation. The phases of this method are followed as discussed next.

- **Specification:** During this phase the core requirements of the ontology are defined . These are to provide support for the following:
  - Representation of sensor measurements pertaining to the use case.
  - Representation of expert knowledge pertaining to the use case.
  - Implementation of Air Quality Index for classification and detection of targeted situations from sensor data.
  - Reasoning and inferencing to determine appropriate control actions to abate unwanted situations.
  - Reuse of relevant terms from existing ontologies.
  - Extensibility to allow incorporation of future components.
- **Knowledge acquisition:** In this phase we acquired knowledge from domain experts in the field of occupational health. We participated in field works in order to acquire domain knowledge and data for evaluation.
- **Conceptualization:** In this phase the structure of the ontology is defined starting with a glossary of terms from domain vocabulary. The structure is aligned with the abstract architecture (see Section 3.1), the problems and their

solutions are described in terms of domain vocabulary.

- **Integration:** Terms from three different ontologies were reused in the ontology for this use case. These are the SSN ontology [43], CSIRO new Time-new ontology [45], and the activity pattern ontology [1].
- **Implementation:** In this phase the ontology was developed using the Protégé ontology editor [109].
- **Evaluation:** Methontology defines evaluation as carrying out technical judgement on the ontology. The ontology is evaluated by populating it with data and querying it for relevant scenarios to show that it can achieve the goals of building it. And by answering the competence questions (see Section 4.2.3)
- **Documentation:** The process, approaches, implementation and usage of the ontology were documented as specification document. The work was also reported in a published paper [4].

Figure 4.3 shows the main concepts and relations of the monitoring layer of the abstract architecture.

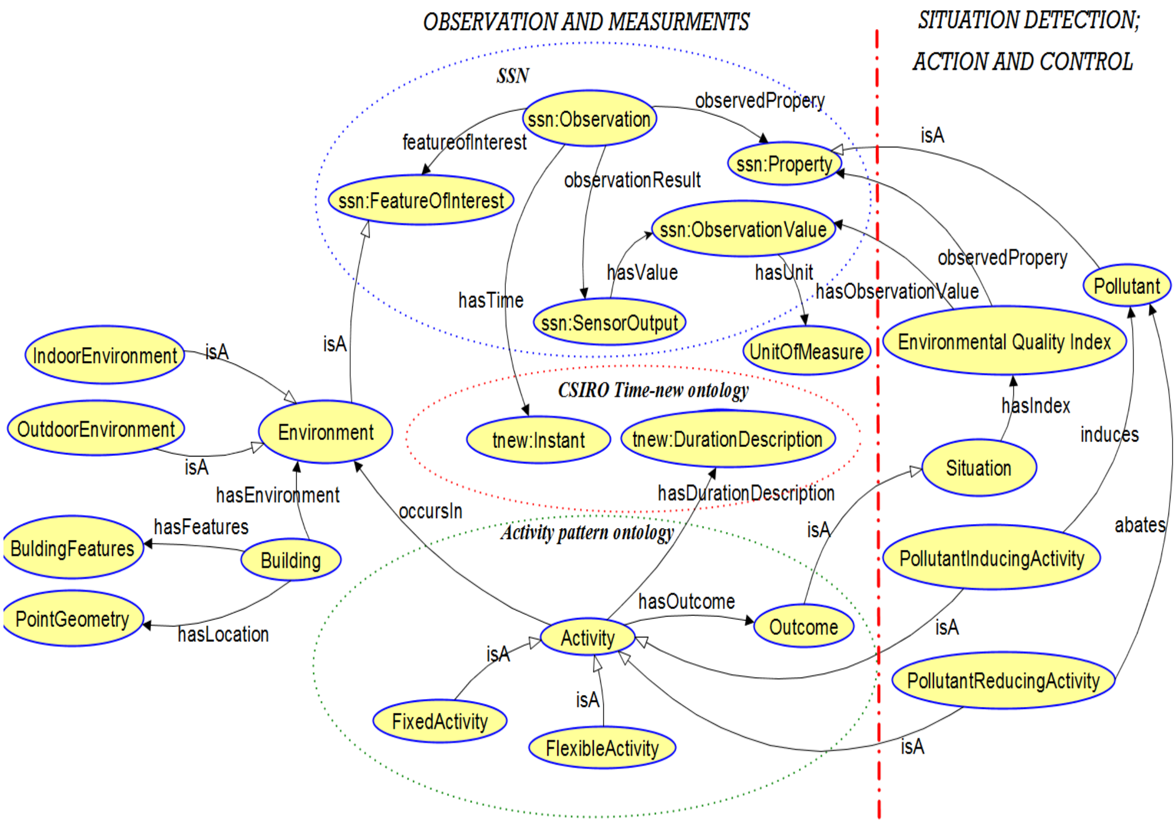


Figure 4.3: Fragment of the IAQ the ontology illustrating the main concepts.

Figure 4.4 shows the fragment of the IAQ ontology representing the Situation analysis layer of the abstract architecture. This includes the Air Quality Index and Thermal Comfort index.

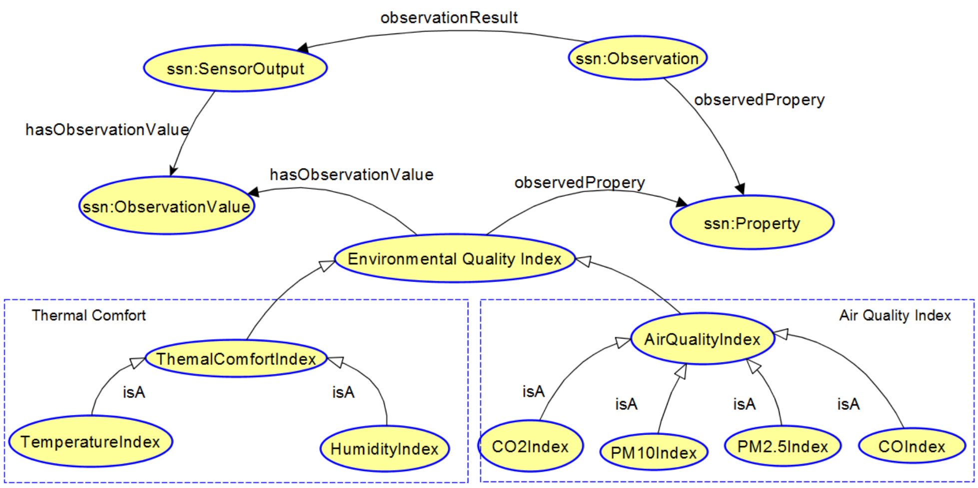


Figure 4.4: Fragment of the IAQ ontology showing terms representing situation detection.

The IAQ monitoring and control system relies on indoor activities of the occupants to control unhealthy levels of the monitored pollutant. Figure 4.5 shows terms in the ontology that represent the action and control layer of the abstract architecture.

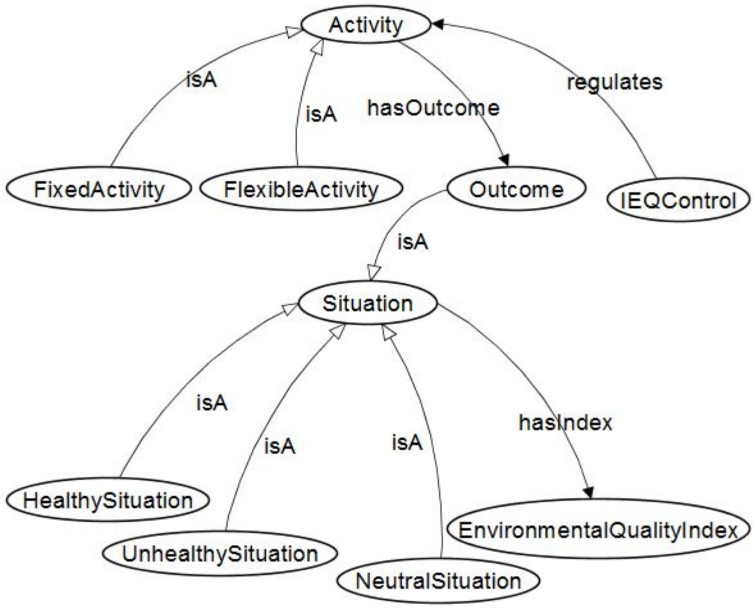


Figure 4.5: Fragment of IAQ the ontology showing terms for situation control.

4.2.5 Implementation of Air Quality Index

The ontology allows for representation and automatic calculation of the Air Quality Index from current sensor observations. Table 4.1 shows the range and categories of the Air Quality Index. It describes six states of health concern, such as good, moderate, unhealthy for sensitive groups, unhealthy, very unhealthy and hazardous.

Table 4.1: PM_{10} Air Quality Index, adapted from Mintz [101]

Air quality	**Ontology classes** (PM_{10} **Index**)	**Concentration** ($\mu g/m^3$)	**Description and control action**
Good	PM10IndexGood	0 - 54	Good Air Quality **Control:** Nil
Moderate	PM10IndexModerate	55 - 154	Moderate Air Quality **Control:** Nil
Unhealthy for Sensitive Groups	PM10IndexUnhealthyfor-SensitiveGroup	155 - 254	Unhealthy air quality for sensitive people **Control:** Limit activities that induce fine particles
Unhealthy	PM10IndexUnhealthy	254 - 354	Unhealthy Coarse Particle Level for all **Control:** Reduce activities that induce fine particles
Very unhealthy	PM10IndexVeryUnhealthy	355 - 424	Very Unhealthy air quality **Control:** Stop {activities that induce fine particles} and Do all {activities that removes fine particles}
Hazardous	PM10IndexHazardous	425 - 604	Hazardous Coarse Particle Level **Control:** Consider evacuating the house - temporarily and consult IAQ expert

Table 4.1 shows the adapted AQI for PM_{10}. Each state corresponds to a range of pollutant concentration values that specifies an index on the AQI. For example , PM_{10} concentrations between 0 and 54 correspond to the Good state. A corresponding instance of PM_{10} Index PM10GoodIndex, represents this state in the ontology. Other states are similarly represented in the ontology. The unhealthy range of values have control feedback messages, which describe what the occupants could do when necessary to abate unhealthy situations detected by the system. Figure 4.6 shows the implementation of the AQI with *Description Logic* (DL) queries on the ontology.

PM_{10} Index	DL Query
PM10IndexGood	(observedProperty value ParticulateMatter10) and (hasValue some double[>= "0.0"^^double , <= "54.0"^^double])
PM10IndexModerate	(observedProperty value ParticulateMatter10) and (hasValue some double[>= "55.0"^^double , <= "154.0"^^double])
PM10IndexUhealthyForSensitiveGroup	(observedProperty value ParticulateMatter10) and (hasValue some double[>= "155.0"^^double , <= "254.0"^^double])
PM10IndexUnhealthy	(observedProperty value ParticulateMatter10) and (hasValue some double[>= "254.0"^^double , <= "354.0"^^double])
PM10IndexVeryUnhealthy	(observedProperty value ParticulateMatter10) and (hasValue some double[>= "355.0"^^double , <= "424.0"^^double])
PM10IndexHazardous	(observedProperty value ParticulateMatter10) and (hasValue some double[>= "425.0"^^double])

Figure 4.6: Implementation of Air Quality Index with DL queries.

The ontology can be queried with SPARQL and DL queries. HermiT 1.3.8.3 reasoner that is bundled with the Protégé ontology editor was used for reasoning on the ontology. It can also be queried in apache Jena, a Java framework for building Semantic Web applications.

4.2.6 Thermal Comfort Index

Similar to the Air Quality Index, a Thermal Comfort Index was implemented using the adopted temperature and humidity values in Table 4.2.

Table 4.2: Temperature and relative humidity scale for thermal comfort

Season	Temperature (^0C)	Relative humidity (%)
Winter	20 - 24	30 - 70
Summer	23 - 26	30 - 70

4.3 Analysis and evaluation

The ontology is evaluated by demonstrating that it can achieve its goals (see Section 4.2.2) and by answering the competency questions (see Section 4.2.3) .

4.3.1 Representation of sensor observations

The first goal of the system is adequate representation of sensor measurements pertaining to the intended use case (indoor air quality). To evaluate the ontology for the capability to represent sensor measurements, it is queried for the sensor data. Query1 show a sample SPARQL query evaluated on the ontology to list the houses and sensor values of locations with unhealthy levels of Particulate Matter (PM_{10}). The result of this query is shown in Table 4.3. This result also answers CQ1 in affirmative.

Query1:
```
PREFIX ieq: <http://www.j.adaptives.cair.ukzn.ac.za/ieq-02#>
SELECT (?loc as ?PM10UnhealthyHouse) ?property (str(?value) as
    ?sensorValue)
WHERE {
        ?x ieq:observedProperty ?property .
        ?x ieq:hasObservationLocation ?loc .
        ?x ieq:hasObservationValue ?value
        FILTER regex(str(?property), 'ParticulateMatter10', 'i'
            )
        FILTER (?value>=254)
}
```

Table 4.3: Result of sensor observations evaluation

PM10UnhealthyHouse	**property**	**sensorValue**
House11	*ParticulateMatter10*	*"255.0"*
House12	*ParticulateMatter10*	*"300.6"*

4.3.2 Situation detection

The second goal of the system is the representation and reasoning service for analyzing sensor data to detect target situations. Figure 4.7 shows PM_{10} observations that are automatically identified as unhealthy by the Air Quality Index query based on sensor observations values. The index was implemented with DL queries (see Section 4.2.5). This result answers CQ2 in affirmative.

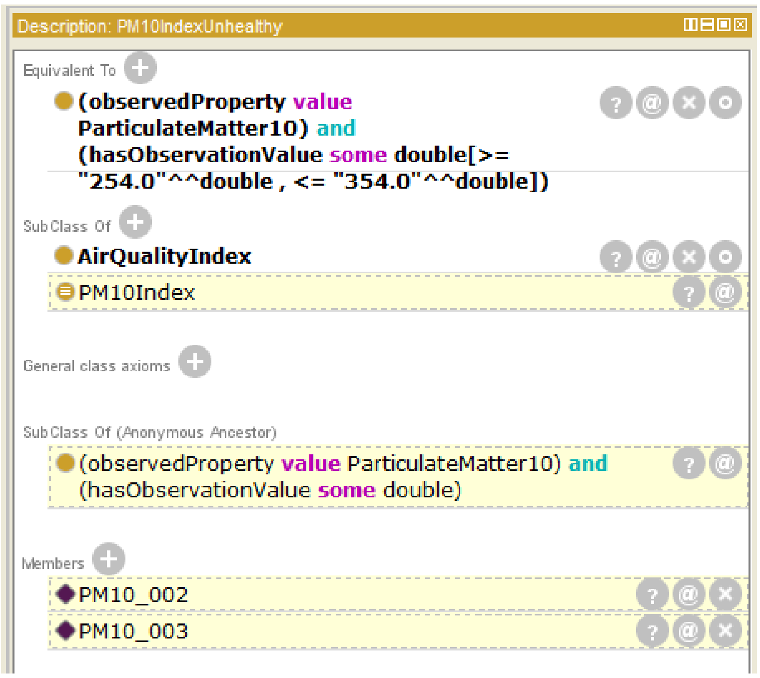


Figure 4.7: Evaluation of situation detection.

4.3.3 Control actions

Control actions are designed as alerts that the system sends via SMS to the house occupants when it is necessary to take actions. To evaluate the capability of the ontology to support this, it is queried for appropriate actions for each of the situation detected in the indoor environment. Query2 shown subsequently is an example query listing for control actions to abate unhealthy PM_{10} pollution. The result of this query is shown in Table 4.4. This result also serves to answer CQ3 in affirmative.

Query2:
```
PREFIX ieq: <http://www.j.adaptives.cair.ukzn.ac.za/ieq-02#>
SELECT (?xs as ?ControlAction)
WHERE {
        FILTER regex(str(?y), 'ParticulateMatter10', 'i')
}
```

Table 4.4: Result of Control Action evaluation query.

ControlAction
useAirFilter
useVacuumCleaner
openWindow

4.4 Summary

This chapter has presented the design of an ontology-driven system for environmental monitoring and control based on the architecture introduced in Chapter 3. The system is evaluated on an indoor air quality use case. The situation analysis layer of the system mainly consists of situation detection based on the ontology-driven Air Quality Index. Quantitative and qualitative states are defined and represented for indoor PM_{10} and thermal comfort, which enables the system to determine the indoor Air Quality Index by reasoning on the ontology. Thermal Comfort Index and Air Quality Index most likely do interact, that is, are not independent of each other. However, this possible interaction is beyond the scope of our thesis. The index used is based on scales found in the literature. However, these can be set to desired scale values. Occupants are alerted by an alarm and feedback messages by SMS to minimize or eliminate pollutant inducing activities when unhealthy levels of the pollutants are detected.

The study in this chapter serves to show how to develop an SSW monitoring application in line with the gap this research is designed to fill. Although the ontology-driven framework presented in this chapter reacts to detected current situations (reactive), the ontology is aimed to drive a proactive application in which the system anticipates and averts the unwanted situation before it happens. Hence the next step towards a proactive framework is to incorporate situation prediction into the system. This will enable it to anticipate the future and produce proactive control actions to avert pending unwanted situations. In the next chapter, we investigate how to incorporate situation prediction in the SSW framework using statistical machine learning techniques.

Chapter 5

Incorporating Statistical Machine Learning in a SSW Framework for Proactive Monitoring and Control

In the previous chapter, an ontology driven system for monitoring and control was proposed. Essentially, the framework in Chapter 4 was able to monitor and only react to situations that have already occurred *(reactive)*. It does not include a mechanism for situation prediction and hence, it could not anticipate future situations. Anticipation is a corner stone of a *proactive* system [146]. To make the framework a proactive one, this chapter presents an approach to incorporate situation prediction in the ontology-driven framework, within the context of SSW. The approach employs statistical machine learning techniques.

Although some recent efforts have proposed predictive reasoning, a semantic method, for predicting future in semantically annotated data streams [87, 92], it is an active research area which is young with new techniques still emerging (see Section 2.5.2). Statistical machine learning provides advanced techniques that support applying machine learning algorithms to learn certain properties and patterns of data to predict future trends. Hence, the choice of statistical machine learning approach for this study.

The study presented in this chapter has been reported in our published paper [5]. The situation prediction for proactive monitoring and control approach is validated on an indoor air quality use case. As mentioned before, indoor air quality is a growing concern [119, 130, 166] and a research area, where proactive monitoring and control in the SSW can be applied. Monitoring and control of Particular Matter of aerodynamic diameter of 2.5 microns ($PM_{2.5}$) in the indoor environment was chosen as the focus of the use case. Most research efforts in indoor air quality have been directed to monitoring concentration levels of indoor pollutants and exposure levels of individuals to the pollutants with applications that react to change in target situations [4, 66, 130]. Such applications allow for responsive actions to situations which have already occurred and are useful for minimizing the effect of these situations. Identifying a possible situation before its occurrence will allow for proactive actions to be taken to avert or enhance its occurrence. A proactive monitoring application in a home can anticipate trends of future pollution levels and trigger control actions to avert the occurrence of such a situation altogether and prevent occupants from exposure to unhealthy levels of pollution.

The main contribution of the study in this chapter is two-fold. The first is the exploration of machine learning for situation prediction from streaming sensor data. This resulted in the selection of a Multilayer Perceptron (MLP) model using a sliding window over the incoming data to predict future values. Secondly, a mechanism for incorporating machine learning models in SSW architectures to support situation prediction is proposed. This is to support taking appropriate control actions ahead of time in order to prevent the occurrence of a future unhealthy situation *(proactive)*. The approach is aimed at combining the high accuracy and performance of statistical predictive techniques and the expressiveness of semantic analytic techniques for proactive monitoring and control applications [5].

This chapter is organized as follows. Section 5.1, presents an overview of the proactive monitoring and control framework. Section 5.2, evaluates the framework with an indoor air quality use case. The incorporation of the predictive model into a stream reasoning framework is presented in Section 5.3. The framework is evaluated in Section 5.4, while Section 5.5 presents a summary of the chapter.

5.1 Proactive monitoring and control in the SSW

5.1.1 Abstract architecture

Figure 5.1, shows the abstract architecture introduced in Chapter 4 and highlights the situation prediction component which is the focus of this chapter. The dotted line highlights the layer while the green line highlights the component focused on in the layer. The abstract architecture emphasizes the use of ontology, a specification approach, for situation detection and machine learning, a learning approach for predicting future situations (see Section 2.5.1, Section 2.5.2). Statistical models can learn from historical data and use the weights generated to analyze current data to predict the future with potentially high precision and sensitivity.

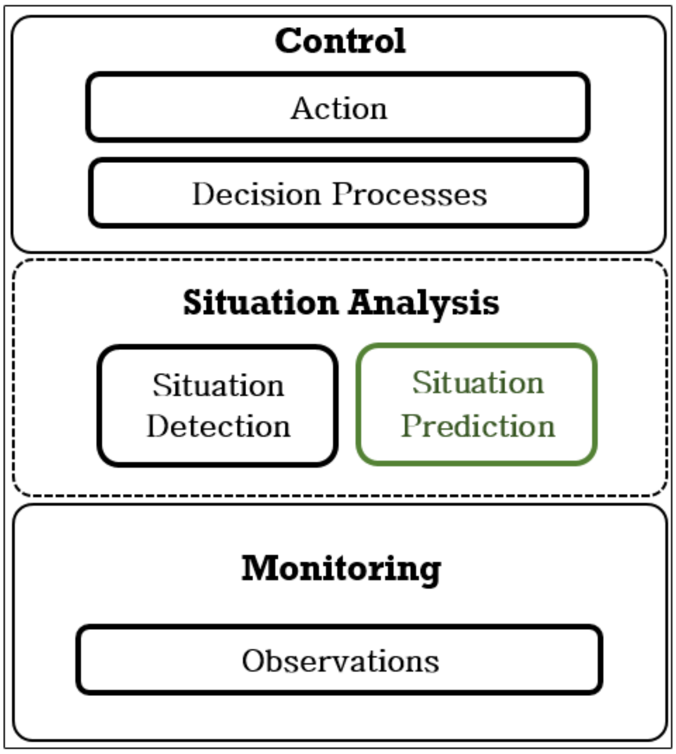


Figure 5.1: Abstract architecture for a proactive SSW framework.

5.1.2 Main components

Figure 5.2 below shows the data flow through the main components of the framework, this is the same as Figure 3.2 in Chapter 3, except that it highlights the components focused on in this chapter (see green border in Figure 5.2). The monitoring layer and situation detection component of the situation analysis layer has been reported in a previous research [4]. The focus of this chapter is on the implementation of the situation prediction process with a statistical machine learning based model and incorporating the outputs of the situation analysis layer of the proposed framework for the decision processing component.

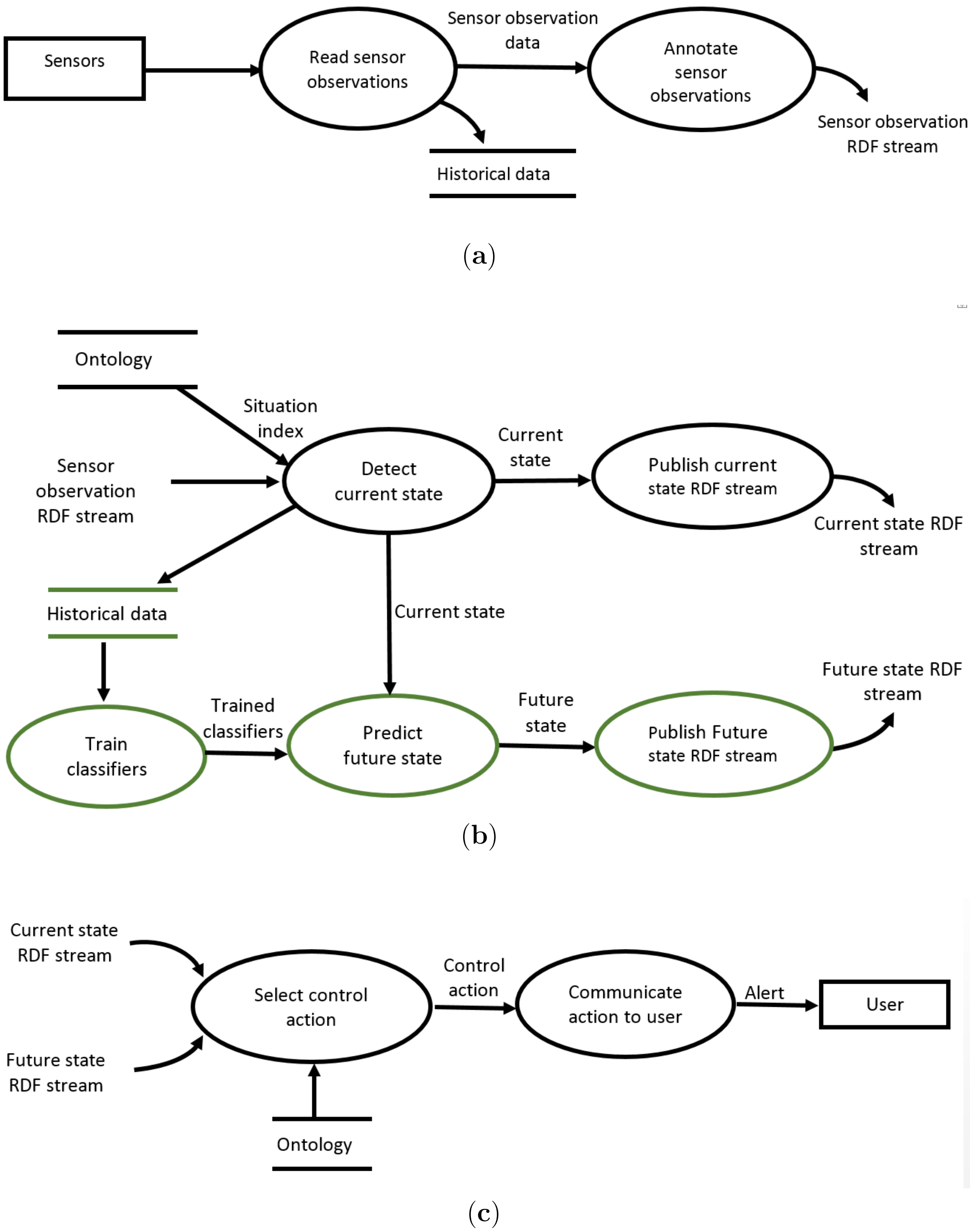


Figure 5.2: Level 1 Dataflow diagram of the proactive SSW framework: (**a**) Monitoring (**b**) Situation analysis (**c**) Control

5.2 Application use case

The use case for this work is an ongoing cohort study [78, 112] along with occupational health researchers investigating the effects of indoor air pollution, especially fine particles pollution on pregnant mothers and children. This is the same use case that was first introduced in Chapter 4, Section 4.1, except that while the *reactive* application in Chapter 4 was focused on monitoring and control of PM_{10}, this chapter is focused on *proactive* monitoring and control of $PM_{2.5}$. Particulate Matters, especially those of the aerodynamic diameter of 2.5 μ or less (also referred to as $PM_{2.5}$), is one of increasingly incriminated indoor pollutants causing life threatening illnesses.

Predicting indoor pollution levels of $PM_{2.5}$ in an indoor environment is a complex and challenging task. The indoor environment is a dynamic and complex system of various environmental phenomena, building features, human activities and infiltrations from the outdoor environment, all of which impact on the fine particles concentration. The proactive monitoring and control system is to predict $PM_{2.5}$ pollution trends effectively and provide proactive control actions for the occupants when necessary to avoid excessive exposure to $PM_{2.5}$ pollution.

The use case area is South Durban. As mentioned before (Chapter 4, Section 4.1), the peculiar characteristics of housing in this community include lack of mechanical heating, ventilation or cooling systems, highly aggravated indoor pollutants through external pollution, and life style choices such as smoking and fossil fuel burning. The area is also in proximity of heavy industries; harmful effects of indoor pollution from outdoor sources have been noted to be more pronounced in residences that are close to heavy industries.

For this research, the goal of the occupational health researcher is to keep the occupants' exposure to particulate pollution within healthy limits. The World Health Organization (WHO) has recommended an exposure limit of $25\mu g/m^3$ daily average for indoor environments [157, 158]. Hence, a Proactive Pollution Monitoring and

Control System is required to monitor and provide control actions to the residents when necessary to avoid exposure to unhealthy $PM_{2.5}$ pollution levels. The system will predict the short term future trend of $PM_{2.5}$ pollution and decide on appropriate control actions to stimulate proactive actions by the occupants to avert exposure to any anticipated unhealthy indoor $PM_{2.5}$ pollution level. The indoor pollution will be controlled via the control of activities of occupants that influence indoor $PM_{2.5}$ pollution. The control action will be communicated as a short message service (SMS) to advise the occupants on proactive actions to take in order to prevent the predicted pollution from occurring. This is an extension of the previous system [4], which only alerts the occupants of unhealthy situations that have already occurred.

Three different houses in the use case area were selected and used for testing the proactive monitoring and control system. One of the locations was first used as a pilot study for a week in April 2015, during the autumn season and the other two were used in October 2015, during the spring season.

5.2.1 Proactive pollution monitoring and control

The proactive pollution monitoring and control system was developed as a prototype implementation of the proposed framework. Sensor units were installed in three houses (Site 1, Site 2 and Site 3; See Figure 5.3). These were implemented with low-cost sensors, mounted on prototyping platforms such as Raspberry Pi to capture and format sensor observation data ($PM_{2.5}$ concentration). The platforms also hosted communication devices to transmit the observation data to the processing server. The sensors sent streaming data over the Internet to the processing server located in the Cognitive and Adaptive Systems Research Laboratory, at the University of KwaZulu-Natal which is 20 km away. Site 1 is about 1.1 km away from Site 2, and about 300 m away from Site 3, while Site 2 and Site 3 are 900 m apart. The processing server hosts the knowledge base, and runs the monitoring and control system.

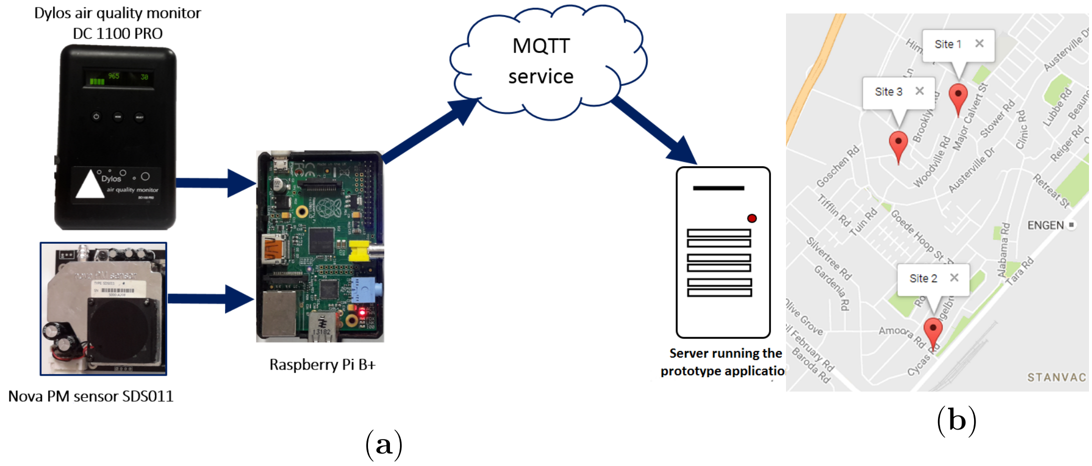


Figure 5.3: (**a**) Main hardware components; (**b**) Google map showing Site 1, Site 2 and Site 3.

The hardware deployed in each site included a sensor network testbed implemented with low cost sensors for the monitored pollutants. $PM_{2.5}$ was monitored with two different low cost sensors, Dylos air quality monitor DC1100 PRO (*Dylos monitor*) and Nova PM sensor SDS011 (*Nova sensor*) [1]. Using two low cost sensors for monitoring simultaneously allows for assuring the quality of the recorded observations. A Raspberry Pi B+ in each location acts as the sensor node that continually transmits streams of sensor observation data to the processing server. The sensor node is equipped with a LB-Link BL-WN151 wireless N adapter that connects to the Internet through HUAWEI E5330 mobile Wi-Fi router and transmits data to the server through a Message Queuing Telemetry Transport (MQTT) service.

The software for the Proactive Pollution Monitoring and Control includes the indoor environmental quality ontology reported on in an earlier study [4], which is now extended with terms to support prediction of future pollution levels and decision rules. The testbed was implemented with Apache Jena framework in Eclipse incorporated development environment. C-SPARQL library, a stream reasoning engine and Apache Jena TDB –a triple store– were also incorporated into the framework.

A predictive model that employs a trained MLP, an Artificial Neural Network model to predict short term pollution levels of $PM_{2.5}$, was implemented for the situation

[1] http://www.dylosproducts.com/dc1100paqmc.html, http://inovafitness.com/en/Laser-PM2-5-Sensor-35.html

prediction component. This was implemented with the Waikato Environment for Knowledge Analysis (WEKA) [69] libraries in the Java environment and incorporated in the architecture. The stream reasoning engine supports incorporating both the current and future $PM_{2.5}$ pollution states to determine appropriate feedback messages. An actuator module is then invoked to send pre-formatted control actions via SMS to the occupant when necessary.

5.2.2 Situation prediction using machine learning

The situation prediction component of the Proactive Pollution Monitoring and Control system aims to predict short term trends of fine particulate matter in the indoor environment. Several factors have been noted to influence $PM_{2.5}$ concentration in the indoor environment, such as indoor and outdoor sources of the particles, fine particles resting on different surfaces can also be resuspended in the air due to impact during activities. Activities, including sweeping, cooking, burning of incense and cigarette smoking are known to influence the concentration of $PM_{2.5}$ captured in the sensor observation data (see Chapter 4).

A sensor data stream is essentially time series data, which requires a time series approach for predicting future values. Prediction of future states can be achieved by pattern *classification* with a *sliding window* technique [108, 149]. Classification is an area of machine learning that involves constructing classifiers for characterizing datasets. A classifier is a function that maps the instances described by a set of attributes to one of a finite set of class labels [60]. Examples of classifiers include Bayesian Network classifiers, Artificial Neural Networks classifiers and Decision Trees classifiers. Classification techniques employ machine learning algorithms to identify and generate a model that fits the relationship between the attribute set and class label of the input data, such that the model can accurately predict class labels of new attribute sets [144]. Situation prediction in this application scenario is treated as *binary classification*. The classifier is made to predict the $PM_{2.5}$ state over a prediction horizon into one of two non-overlapping classes (*"Good"* or *"Poor"*) guided by the WHO recommended exposure limits for indoor $PM_{2.5}$ [157].

The sliding window approach for classification on time series data was adopted to predict $PM_{2.5}$ short term pollution levels 30 minutes (m) and 1 hour (h) into the future. A *sliding window* is a fixed length of data that slides through the temporally ordered data stream. Sliding windows can be useful for two main purposes in time series data classification tasks. First, to select a fixed size of the most recent attributes from the evolving time series data as input for the classifier for predictions. Second, to slide through historical data and select a fixed size of data to update the classifier. In our approach, a sliding window is used to select attributes for generating feature-sets for the classifier to make predictions. Five different classifiers were considered for predicting $PM_{2.5}$ short term pollution levels in this study. These are discussed below.

- *Bayesian Network (BN):* BN also referred to as *belief network* is an annotated directed acyclic graph that support representation of joint probability distribution over a set of random variables. A vertex in the graph represents a random variable while the edges represent dependencies between the variables. A conditional probability table is maintained at each node. A BN classifier can learn appropriate Bayesian Network structure, and the probability tables from training data given the class variable. Classification is done based on joint probability distributions over class variables, given the particular instance of input variables. A class label with the highest posterior probability is predicted [60]. BayesNet is an implementation of BN in WEKA library [159].

- *Multilayer Perceptron (MLP):* MLP is one of a family of computation models called Artificial Neural Networks (ANN). They are used in machine learning and cognitive science to emulate the biological nervous system in computing functions. An ANN consists of several interconnected 'neurons' and is capable of changing its internal structure based on the data that flows through it either from external or internal source. ANNs have been found notably suitable for non-linear classification tasks. A MLP consists of three type of layers: the input layer, one or more hidden layers and the output layer. MLP classifiers have been widely and successfully used for time series prediction tasks [152].

- *Decision Table (DT):* DT is a rule based classifier which functions in the form of a look up table. DT consists of hierarchical tables such that each entry in a higher level table is broken down by the values of a pair of additional attributes to form another table, a process called decomposition. As such, DT has two components, a list of attributes also called a schema, and a multiset of labeled instances referred to as the body [89]. A decision table algorithm generates decision tables from training data for a specific prediction task. Given unseen input data, the generated table is searched for the class label whose attributes are the same as that of the unseen data to determine the class label for the data. In the case of no class has exactly same attributes, one with very similar attributes measured by some metrics (nearest neighbor) is predicted [97].

- *J48:* This is an open source Java implementation of C4.5, a decision tree method. A decision tree classification algorithm builds decision trees from labeled input datasets. A non-leaf node on the tree represents an attribute variable, while leaf nodes represents class variables. The J48 classifier implements a concept referred to as information gain, an information theoretic concept which is used to measure the amount of information an attribute set contains. The algorithm uses the information gain to generate rules for assigning class labels to unseen data [83].

- *Random Forests (RF):* RF is an ensemble learning method. Ensembles are methods that implement several classifiers and aggregates their results. RF employs a method called bagging to aggregate results from several decision tree classifiers. Successive trees in bagging are independently constructed using a bootstrap sample of the dataset, such that a simple majority vote is taken on the result of the trees to make a prediction [30, 94]. RF has been noted to give good performance on time series data [82, 170].

The data set consists of time series data of $PM_{2.5}$ concentration level generated from the sensor observation data. One week of continuous $PM_{2.5}$ sensor observation data of one minute resolution was collected from each site for this study. The data was captured by two sensors, the Dylos monitor and the Nova sensor. Dylos monitor records $PM_{2.5}$ observations in particle counts per cubic feet, while the Nova sensor records observation in micrograms per cubic meter (μg/m^3). Conversion of the data from Dylos monitor to μg/m^3 was achieved using the widely used method derived by Semple et al. [132, 141].

Sensor data from low cost sensors can be inherently noisy. Hence, to minimize the noise in the data, a 30 m simple moving average of the actual 1 m resolution sensor observation data is used for the analysis. The sliding window technique maintains a queue of constant length in the form of *first in first out* (FIFO) with one minute resolution sensor observation data. At every minute a new sequence is formed which differs from the previous sequence only by addition of the newest time step observation data, and removal of the oldest time step observation data in the sequence. More formally, if O_t represents the observation at current time t, at every time step, a new sequence consisting of a series of n observations is formed by pushing-in the new observation as O_t and popping out the oldest observation $O_{t-(n-1)}$ from the previous sequence.

The features for building the classifiers include timestamps, mean of the sliding window sequence, class value for the mean, and class label for the target class. The class value and class label are categorical and binary, that is, two non overlapping classes (*"Good"* and *"Poor"*). Guided by the WHO recommended exposure limits to indoor $PM_{2.5}$ [157], concentration values that are less than or equal to 25 μg/m^3 are set to *"Good"* and those that are greater than 25 μg/m^3 are set to *"Poor"* (see Table 5.1).

Table 5.1: Class values, adapted from WHO recommended exposure limits for indoor $PM_{2.5}$ [157].

$\mathbf{PM_{2.5}}$ Concentration (μg/m^3)	Class Value
≤ 25	*"Good"*
> 25	*"Poor"*

5.2.3 Experiments

Experiments were carried out with time series techniques such as Auto Regressive Integrated Moving Average (ARIMA) but yielded no satisfying result for this use case. Hence, the adoption of a sliding window technique. 6480 data points of one m resolution, this consists of four and a half day continuous observation data that was selected from each site data for analysis (see Section 5.2.2). The data was analyzed to select the appropriate machine learning algorithm for the use case and to determine the optimal training methods for the model. The experiments are described below:

Experiment 1: Data visualization

The aim of the data visualization is to visualize the data from each site and understand class distribution of the data. First, the one minute resolution raw observation data from both Dylos monitors and Nova sensors were plotted together in line charts to show the trends of $PM_{2.5}$ in the sites and also to see the agreement between the two sensors. Second, 30 m moving average data from both sensors was also plotted.

Figure 5.4 shows the visualization of the raw $PM_{2.5}$ observations from the sites. The data captured by the Nova sensor is less accurate than Dylos monitor observations (see Figure 5.4), therefore, data captured by the Dylos monitor is used for the remaining experiments. The figure shows Site 1 to be a heavily polluted house. This corresponds to the characteristics of the house; highly congested with one of the windows perpetually opened. Site 2 and Site 3 are much less polluted, they are cleaner and less congested. The high frequency of class *"Poor"* in Site 1 may also be due to seasonal variations, since Site 1 data was collected in April during the autumn season and data from the other two sites was collected in October during the spring season.

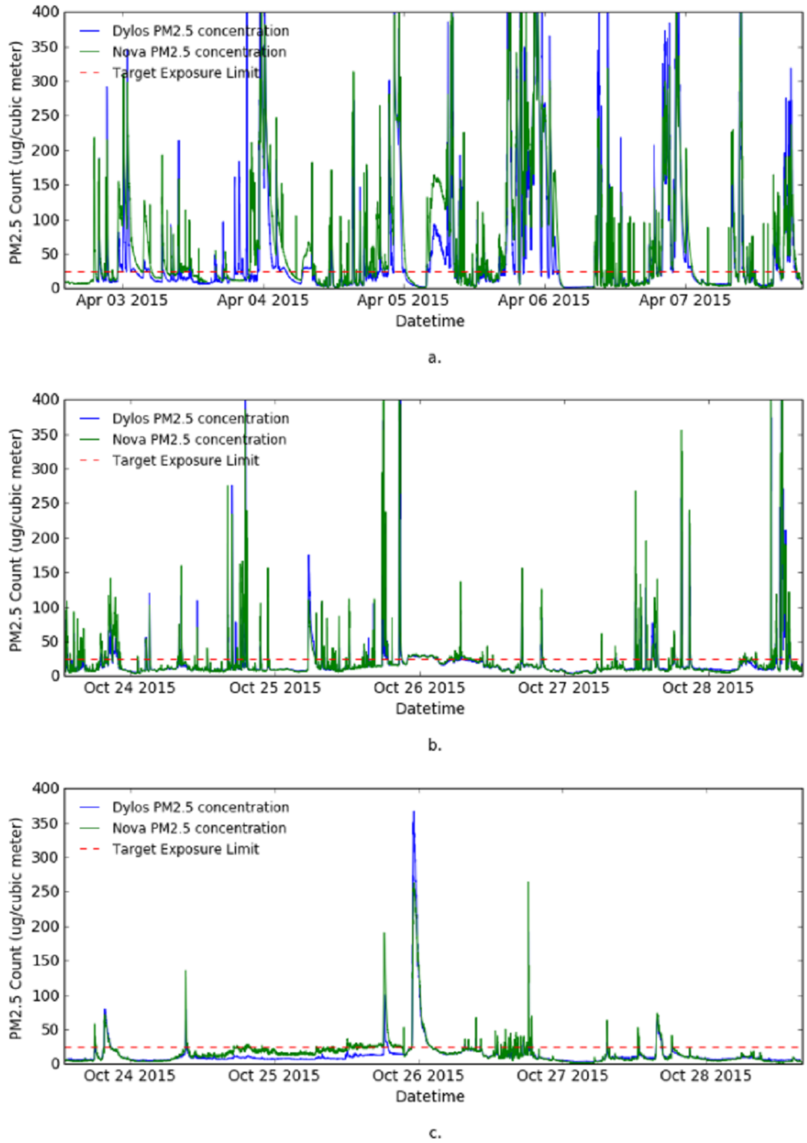


Figure 5.4: Line graph of raw sensor observation from the monitored sites. (**a**) Site 1; (**b**) Site 2; and (**c**) Site 3.

Figure 5.5 shows the 30 m simple moving average of observation data from the three sites and the target exposure limit for $PM_{2.5}$. From the graph, Site 1 is identified to fall in the category of the houses targeted for the Proactive Pollution Monitoring and Control system.

As a result of the visualization experiments, Site 1 is identified to fall into the category of the houses whose occupants are at risk of excessive exposure to fine particle pollution. Hence, the remaining experiments are performed on the data from Site 1, captured with the Dylos monitor.

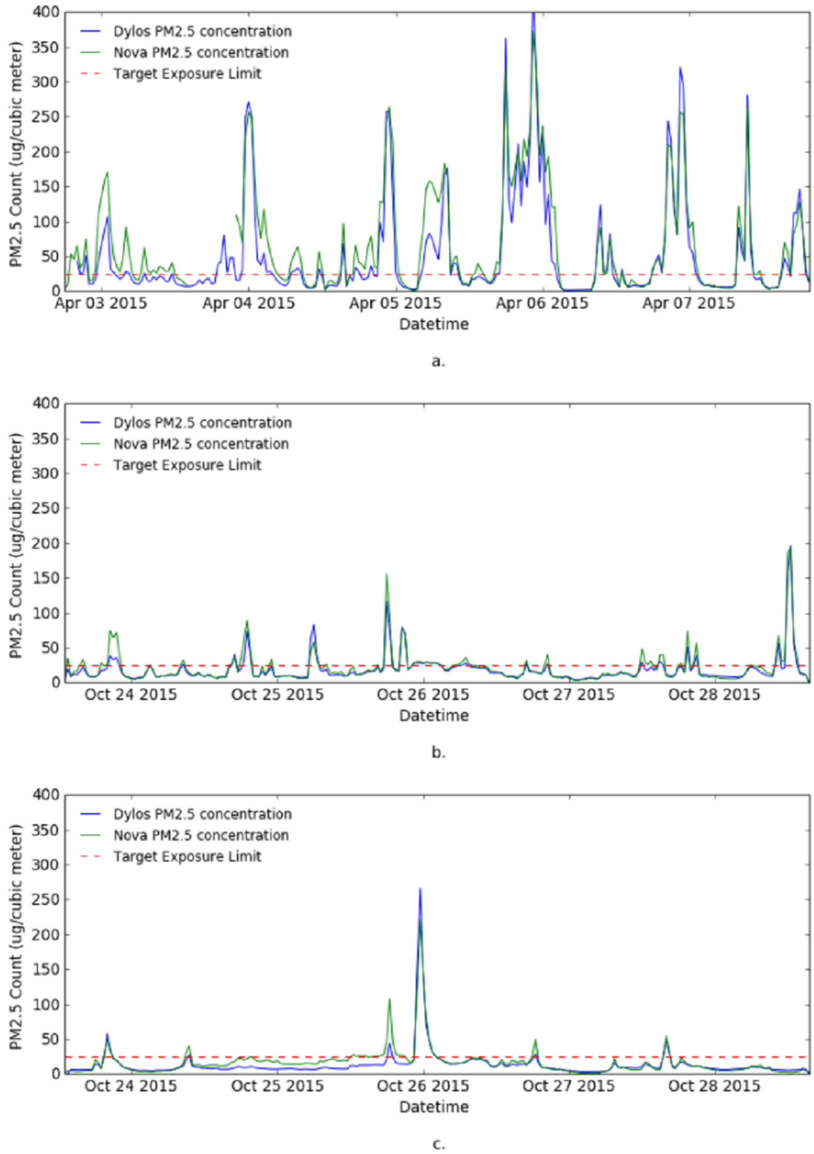


Figure 5.5: Line charts showing 1 min data from the monitored sites. (**a**) Site 1; (**b**) Site 2; and (**c**) Site 3.

Experiment 2: Evaluation of classifiers for predictive modeling

The aim of Experiment 2 is to select the appropriate classifier for a short term prediction of $PM_{2.5}$ in the indoor environment. This experiment simulates the real live use case of the predictive model. For this experiment, the 1 m resolution data was further resampled to 30 m resolution such that a data point represents an average of sensor observation for the past 30 m. Resampling to 30 m resolution makes the prediction task over a 30 m horizon a one time-step prediction. The 30 m resolution data is used to generate input data for the classifiers in this experiment. The dataset is partitioned to allow for the classifiers to slide through the entire dataset at 6 h time-steps.

First, the model initializes by training the classifiers with the first 36 h observation data, then the classifier is made to predict target labels of unseen data for the following 6 h. After the prediction, the 6 h of unseen data is added to the training data and the classifier is retrained. This process is repeated through the entire dataset. All the classifiers were evaluated through the dataset in this manner. Table 5.2 shows the partitioning of the dataset for this experiment. The partitions on the left side of the table are for the training datasets while those on the right side are for the test datasets.

Table 5.2: Dataset partitions for evaluating classifiers.

Training Set		**Train Set Size**	**Testing Set**		**Test Set Size**
From	**To**		**From**	**To**	
3 April 2015 10:00	4 April 2015 21:30	72	4 April 2015 22:00	5 April 2015 3:30	12
3 April 2015 10:00	5 April 2015 3:30	84	5 April 2015 4:00	5 April 2015 9:30	12
3 April 2015 10:00	5 April 2015 9:30	96	5 April 2015 10:00	5 April 2015 15:30	12
3 April 2015 10:00	5 April 2015 15:30	108	5 April 2015 16:00	5 April 2015 21:30	12
3 April 2015 10:00	5 April 2015 21:30	120	5 April 2015 22:00	6 April 2015 3:30	12
3 April 2015 10:00	6 April 2015 3:30	132	6 April 2015 4:00	6 April 2015 9:30	12
3 April 2015 10:00	6 April 2015 9:30	144	6 April 2015 10:00	6 April 2015 15:30	12
3 April 2015 10:00	6 April 2015 15:30	156	6 April 2015 16:00	6 April 2015 21:30	12
3 April 2015 10:00	6 April 2015 21:30	168	6 April 2015 22:00	7 April 2015 3:30	12
3 April 2015 10:00	7 April 2015 3:30	180	7 April 2015 4:00	7 April 2015 9:30	12
3 April 2015 10:00	7 April 2015 9:30	192	7 April 2015 10:00	7 April 2015 15:30	12

Two different classifiers were constructed and evaluated for each of the five different classification methods selected. The first classifier is trained to predict for the half hour horizon and the second classifier is trained to predict for one hour horizon.

Evaluation Criteria: In order to evaluate the performance of selected classifiers, a *confusion matrix* was constructed from the results of the classification, and the widely accepted metrics for binary classification tasks in machine learning community which include *Accuracy*, *Precision*, *Recall (Sensitivity)*, *Specificity* and *F-Measure* [120, 138], were calculated from the confusion matrix. This classification task is focused on identifying the classifier that can better predict the *"Poor"* classes in the dataset over the prediction horizon. Therefore, when a *"Poor"* state is correctly classified as *"Poor"*, it is regarded as *true positive* (TP), and when a *"Good"* state is correctly classified as *"Good"*, it is regarded as *true negative* (TN). Likewise, a *"Good"* state wrongly classified as *"Poor"* is *false positive* (FP) and a *"Poor"* state wrongly classified as *"Good"* is *false negative* (FN). The counts of TP, TN, FP and FN predicted by the classifier is used to generate the confusion matrix (see Table 5.3) and the evaluation metrics as discussed below.

Table 5.3: Confusion matrix.

Actual Class Value	**Classified as *"Poor"***	**Classified as *"Good"***
"Poor"	TP	FN
"Good"	FP	TN

- *Accuracy:* Accuracy represents the overall performance of the classifier and it denotes the proportion of the whole testset (TP + FP + TN + FN) that are correctly classified (TP + TN) [138].

- *Precision:* Precision also referred to as *confidence* in the data mining community [120] denotes the proportion of predicted positive cases that are actually positive (*"Poor"*) in reality.

- *Sensitivity:* This is otherwise known as *recall* and it evaluates the proportion of the real positive states that are predicted positive [120].

- *Specificity:* Specificity or true negative rate is an inverse of recall, which denotes the proportion of real negative cases (*"Good"*) that are correctly predicted negative [138].

- *F-Measure:* F-Measure is an harmonic mean which combines precision and recall [120, 138].

Result: Table 5.4 presents the result of the evaluation on the classifiers for predictive modeling. Most of the classifiers show good precision and classification accuracy; however, for the analysis, we are focused on not only precision but also on the balance between how sensitive the classifier is to the *"Poor"* states and how much it recognizes the *"Good"* classes (specificity). The Random Forests classifier demonstrated the highest precision of 0.906 for the half hour prediction horizon but has the least sensitivity (0.774). This is evident in the bias to the *"Good"* classes observed in the prediction task. The BN and the MLP demonstrate best performance in predicting $PM_{2.5}$ states for the half hour horizon (see bold figures in Table 5.4), but the BN demonstrates lesser precision in predicting states for the one hour horizon. As a result of this experiment, MLP was chosen to model this use case.

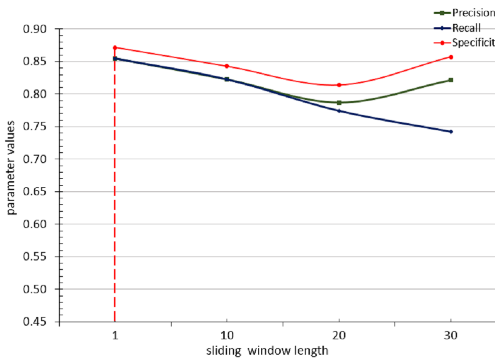
Table 5.4: Precision, sensitivity, specificity and F-Measure of evaluated classifiers on the Site 1 dataset.

Prediction Horizon	**Classifier**	**Accuracy**	**Precision**	**Sensitivity**	**Specificity**	**F-Measure**
30 m	BN	0.864	**0.855**	**0.855**	**0.871**	0.855
	DT	0.856	0.864	0.823	0.886	0.843
	J48	0.856	0.852	0.839	0.871	0.846
	MLP	0.864	**0.855**	**0.855**	**0.871**	0.855
	RF	0.856	0.906	0.774	0.929	0.835
1 h	BN	0.780	**0.758**	**0.770**	**0.789**	0.764
	DT	0.773	0.804	0.672	0.859	0.732
	J48	0.773	0.816	0.656	0.873	0.727
	MLP	0.788	**0.780**	**0.754**	**0.817**	0.767
	RF	0.758	0.822	0.607	0.887	0.698

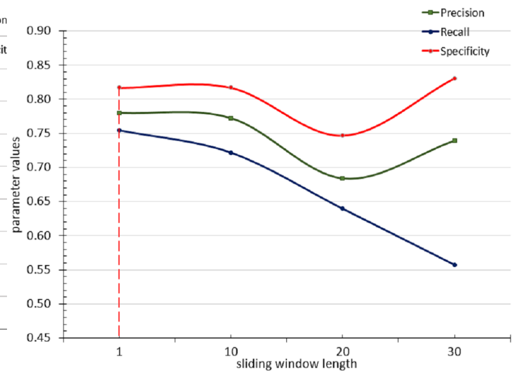
Experiment 3: Evaluation of sliding window sizes

This experiment aims to determine the optimal sliding window length for training the MLP that was selected for this study in Experiment 2. MLP classifiers were evaluated on four different datasets, each of which were prepared with different sliding window lengths ($n = 1$, $n = 10$, $n = 20$ and $n = 30$) and partitioned as shown in Table 5.2. The classifiers were made to predict next class values for both 30 min and 1 h prediction horizons. The performance of the classifiers in terms of precision, recall and specificity on each of set of the data was plotted in line charts.

Result: Figure 5.6 shows the result of this experiment. It reveals that increasing the sliding window lengths of input data to the classifiers steadily decreases the performance of the classifiers in predicting the target classes. The point at which specificity and precision starts increasing when sensitivity (recall) keeps decreasing demonstrates a point where bias towards one of the target classes (*"Good"*) sets in, and starts increasing. That is, the model steadily loses sensitivity to the *“Poor”* class from this point. The dataset with window length $n = 1$ gave the best performance (highlighted with dotted vertical lines in Figure 5.6). Sensitivity especially demonstrates a free fall with the increase in sliding windows length. This observation may be due to the notion that more recent data is more relevant to the future than older ones [110]. A more detailed tabulated result of this experiment is presented in Table 5.5.



(**a**)



(**b**)

Figure 5.6: Line charts showing precision, recall and specificity against different sliding window length, (**a**) half hour prediction horizon; (**b**) one hour prediction horizon.

Table 5.5: Accuracy, precision, recall, specificity and F-measure of MLP classifiers in dataset with different sliding windows.

	Sliding Window Length	**Accuracy**	**Precision**	**Recall**	**Specificity**	**F-Measure**
	1	0.864	0.855	0.855	0.871	0.855
30 min	10	0.833	0.823	0.823	0.843	0.823
	20	0.795	0.787	0.774	0.814	0.780
	30	0.803	0.821	0.742	0.857	0.780
	1	0.788	0.780	0.754	0.817	0.767
1 h	10	0.773	0.772	0.721	0.817	0.746
	20	0.697	0.684	0.639	0.746	0.661
	30	0.705	0.739	0.557	0.831	0.636

5.3 Incorporation of the predictive model in the framework

The selected MLP predictive classifier was incorporated into the system using the WEKA library in Eclipse, a Java based Itegrated Development Environment. The situation prediction component consists of two different MLP classifiers to achieve two different horizons of prediction. The first was trained to predict pollution levels for the next half hour, and the second for the next one hour. The result of the situation prediction generated from the models is incorporated into the stream reasoning framework by encoding it as Resource Description Framework (RDF) triples (see Figure 5.7). The C-SPARQL RDF stream reasoning engine supports registered queries to combine RDF streams and static RDF triples (in ontologies) for reasoning. Through this process, the RDF streams of predicted $PM_{2.5}$ pollution trends which correspond to the future situation of the indoor air quality is combined with RDF streams of the current situation detected by the Air Quality Index for decision processing.

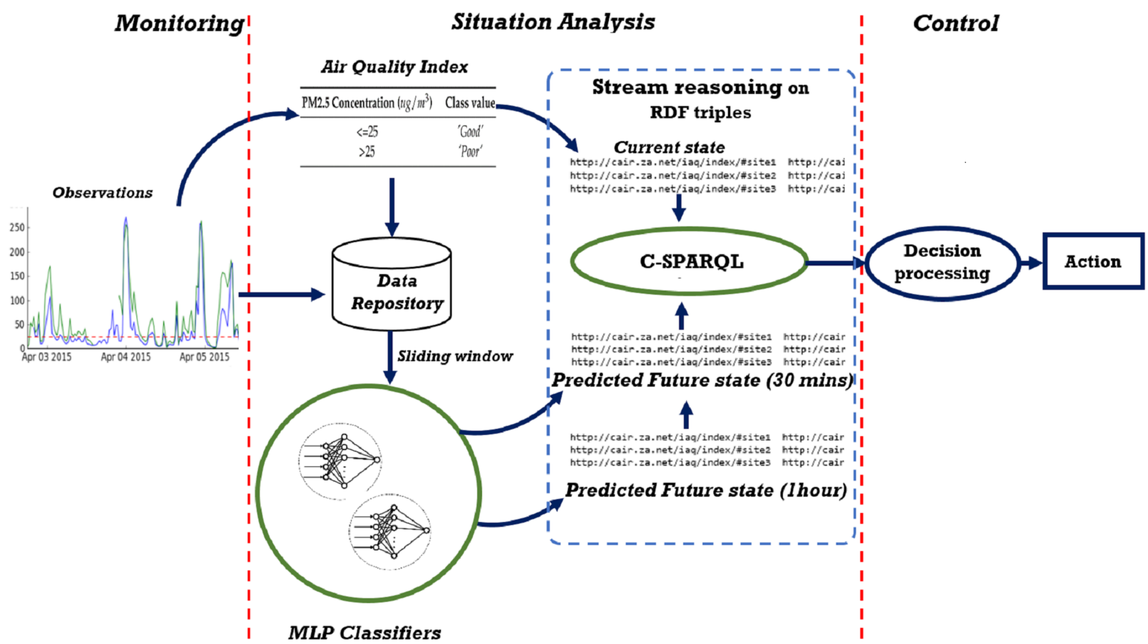


Figure 5.7: Example implementation of the proposed framework.

Three continuous queries are registered with the C-SPARQL engine to filter the RDF streams at the current time (as indicated by the Air Quality Index module) at the next half hour and at the next one hour. In order to be unobstructive, the system does nothing when the air quality is *"Good"*. At any time that either the current state or the predicted state is *"Poor"*, the decision processing module in the control layer is notified. The values detected by the monitoring queries are recorded in the ontology for reasoning by the decision processing module. Figure 5.8 shows a fragment of the ontology illustrating how an observation is stored. The model is based on the SSN ontology [43].

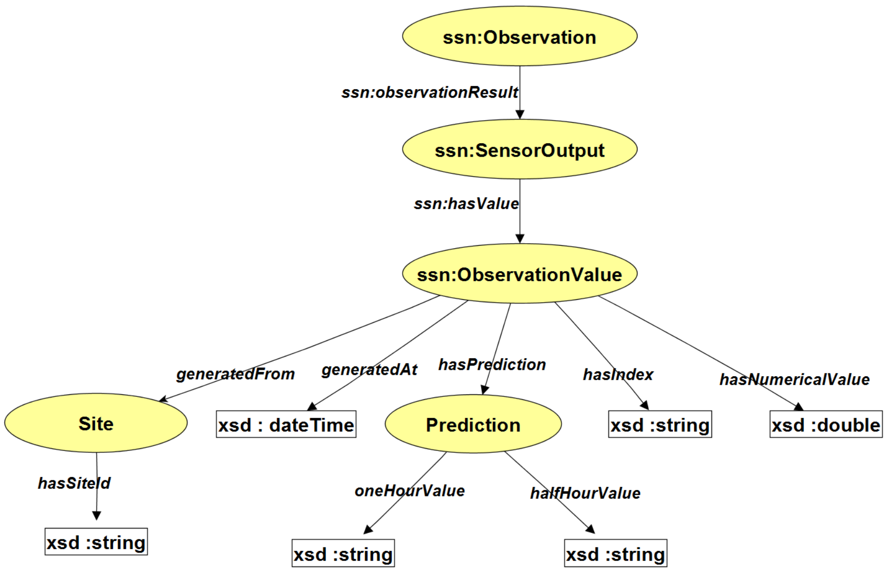


Figure 5.8: Fragment of the ontology showing the data model.

The following listings illustrate how triples are stored in the ontology, and how they can be processed for monitoring and control with continuous queries. We use *iaq-owl* as a shorthand notation for the Internationalized Resource Identifier (IRI).

- *How data is stored in the ontology*

```
iaq-owl:SEQ2500 iaq-owl:generatedFrom iaq-owl:site01
iaq-owl:SEQ2500 iaq-owl:generatedAt "01:25:12.100"^^xsd:time
iaq-owl:SEQ2500 iaq-owl:hasPrediction iaq-owl:PRE7900
iaq-owl:SEQ2500 iaq-owl:hasIndex "good"^^xsd:string
iaq-owl:PRE7900 iaq-owl:halfHourValue "poor"^^xsd:string
iaq-owl:PRE7900 iaq-owl:oneHourValue "poor"^^xsd:string
```

- *Monitoring current air quality state*

  This query continually filters through the indoor Air Quality Index stream to notify the decision processing component about the current air quality detected by the index.

```
REGISTER QUERY CurrentStateQuery
AS PREFIX iaq-owl: <http://iaq-ukzn.ac.za/iaq.owl#>
SELECT ?site ?current ?t
FROM STREAM <http://iaq-ukzn.ac.za/iaqindex/stream>
 [RANGE 10m STEP 10m]
WHERE {?seq iaq-owl:hasSeqID ?sid
         ?sid iaq-owl:generatedFrom ?site.
         ?sid iaq-owl:generatedAt ?t.
         ?pid iaq-owl:hasIndex ?current.
         FILTER (?current = "good"^^xsd:string)}
```

- *Monitoring half hour prediction state:*

  This query monitors the predictions over a 30 m horizon, it is activated to notify the decision processing component when air quality predicted in the next 30 m is *"Poor"*.

```
REGISTER QUERY halfHourPredictionQuery
AS PREFIX iaq-owl: <http://iaq-ukzn.ac.za/iaq.owl#>
SELECT ?site ?p1 ?t
FROM STREAM <http://iaq-ukzn.ac.za/prediction/stream>
 [RANGE 10m STEP 10m]
WHERE {?seq iaq-owl:hasSeqID    ?sid
         ?seq iaq-owl:generatedFrom ?site.
         ?sid iaq-owl:generatedAt ?t.
         ?sid iaq-owl:hasPrediction ?p.
         ?p iaq-owl:halfHourValue ?p1.
         FILTER (?p1 = "poor"^^xsd:string)}
```

- *Monitoring one hour prediction state*

  This query is activated to notify the decision processing component when air quality predicted in the next one hour is *"Poor"*.

```
REGISTER QUERY oneHourPredictionQuery
AS PREFIX iaq-owl: <http://iaq-ukzn.ac.za/iaq.owl#>
SELECT ?site ?p2 ?t
FROM STREAM <http://iaq-ukzn.ac.za/prediction/stream>
 [RANGE 10m STEP 10m]
WHERE {?seq iaq-owl:hasSeqID    ?sid
        ?seq iaq-owl:generatedFrom ?site.
        ?sid iaq-owl:generatedAt ?t.
        ?sid iaq-owl:hasPrediction ?p.
        ?p iaq-owl:oneHourValue ?p2.
        FILTER (?p2 = "poor"^^xsd:string)}
```

RANGE and *STEP* are operators used in C-SPARQL queries to support time windows. *RANGE* specifies the size of the time window that the query filters through, while *STEP* specifies time steps with which the time window slides forward. Setting both *RANGE* and *STEP* to the same value (for example 10 min as used in this use case) specifies a tumbling window scenario, in which the time window does not slide, but rather, at the end of a time window, another time window starts in a tumbling manner. This means that subsequent results do not contain observations from previous results. In this example, the window's size is set to 10 min, but this can be set as desired.

The state values detected by the continuous queries can be used by the decision processing component for reasoning with decision rules in the ontology in order to determine the appropriate actions at a point in time. For example, lets consider as a target situation when the predicted $PM_{2.5}$ state is persistently *"Poor"* for up to thirty minutes. We can represent this in the system as when both the half hour and one hour prediction results are *"Poor"*. In this situation, the Proactive Pollution Monitoring and Control System needs to warn occupants to take some recommended proactive actions to avoid the predicted situation. The listing below demonstrates reasoning-logic by the decision processing component in this example.

```
house(?site), sequence(?sid),
generatedFrom(?sid,?site),
```

```
hasIndex(?sid, !"poor"),
hasPrediction(?sid, ?pid),
halfHourValue(?pid, "poor")
oneHourValue(?pid, "poor"),
-> PM25pollutionPredicted(?site, ?true)
```

The decision rule can be implemented in any reasoning infrastructure that is compatible with the Semantic Web, such as Semantic Web Rule Language (SWRL), SPARQL or the JENA rule engine [2]). In the use case scenario, when the pollution is predicted, the decision manager can activate the actuation module to send an appropriate control action to the occupants in order to prevent the pending unhealthy situations from happening. An example of this could be: "Alert: *Unhealthy Fine Particle Level predicted soon*; Proactive Control Advice: *Please avoid smoking, burning incense and excessive cooking indoors*". More details about using activities to control indoor $PM_{2.5}$ pollution is presented in one of our papers [4].

5.4 Analysis and evaluation

In order to determine how the Proactive Pollution Monitoring and Control System will perform in the field, we carried out evaluation tests based on the test data used to evaluate the classifiers (see Section 5.2.3). The test data for the evaluation consists of 132 observations in all (see Section 5.2.3). The data was made to run through the components of the system. The performance of the components and the overall efficiency of the system were analyzed. The system used for the evaluation is an ASUS laptop running Windows 7, with Corei5 (Intel(R) Core(TM)i5-3337U CPU @1.80GHz) processor and 12.0 GB installed memory. Result of the analysis and evaluation of the system with respect to design decisions made on each of the components are discussed below.

The situation prediction component initializes by training the classifiers (see Section

[2]www.w3.org/Submission/SWRL/, https://jena.apache.org/documentation/inference/#rules

5.2.2) with 36 h of historical data (see Section 5.2.3). Over ten runs, the average initialization time was 39,208.0 ms (≈0.65 min) to train MLP classifiers for the half hour prediction and 47,098.4 ms (≈0.78 min) to train the classifiers for the one hour prediction. The classifiers then effectively processed each subsequent prediction task in a maximum of 1 ms in all the cases. However, the system is also designed to update the classifiers every 6 h with the most recent data. We compared the training times of the MLP classifiers with that of BN classifiers which was found equally suitable for this work (see Section 5.2.3). Table 5.6 shows the variation of training time as the size of datasets grows. The re-training time for MLP classifiers increases rapidly as the dataset grows, while the re-training time of BN is minimal and remains relatively constant after the initialization. This experiment reveals that although the MLP model has a slightly better predictive performance than the BN in this study, it is not as scalable as the BN. Hence, the choice of MLP over BN for the system is a trade-off between the predictive performance and scalability. Given the poor model update speed of the MLP as the data set grows, the BN is a more likely choice for implementation. However, further investigation is required on mechanisms for reducing the model update time for the MLP.

Table 5.6: Performance of situation prediction classifiers during updates.

	MLP		**BN**	
Dataset Size	**One Hour Classifier Training Time (ms)**	**Half Hour Classifier Training Time (ms)**	**One Hour Classifier Training Time (ms)**	**Half Hour Classifier Training Time (ms)**
72	39,208.0	47,098.4	288.2	301.2
84	45,280.4	49,701.2	4.8	3.7
96	54,979.6	55,967.2	4.2	3.0
108	64,487.0	62,518.6	3.4	3.7
120	74,083.8	65,513.8	3.4	3.1
132	76,995.2	71,036.4	3.2	2.9
144	85,660.4	81,994.4	3.0	3.5
156	92,779.4	91,990.8	3.2	4.1
168	92,376.6	97,852.6	4.2	7.0
180	97,573.0	104,304.6	3.8	3.3
192	109,404.8	111,131.4	4.8	4.5

The situation detection component, which detects the current situation by interpreting observation data based on the Air Quality Index (see Section 5.2.2) identifies all the situations correctly. The output of this component also serves directly as labeled data for retraining the classifiers during system updates.

Stream reasoning with C-SPARQL is used to monitor three different streams (see Section 5.3) in the system, namely, the current pollution situation, the half hour prediction and the one hour predictions. Out of the 132 observations in the test data, 62 observations have either half hour predictions or one hour predictions that are "*Poor*". The queries effectively detected all the targeted situations correctly.

The decision to activate alarms is based on the result of a SPARQL query that is evaluated on the ontology at specified intervals, which was set to 10 m for the purpose of this evaluation. The query filters through the data to detect situations in which half hour and one hour predictions are both "*Poor*" for the past 10 m in order to activate control actions. When C-SPARQL is used to filter the predictions, only the 62 triples that have either half hour prediction or one hour predictions as "*Poor*" were recorded in the ontology. In a query test that was repeated ten times, the average execution time of SPARQL query was found to be 295 ms. We compared this with the execution time of SPARQL query when all the observations were streamed into the ontology, that is, when C-SPARQL is not used. In this case, the dataset in the ontology includes the triples representing the predictions of all the 132 observations. The average execution time of the query is 441 ms. The difference of 146 ms may seems little because of the minimal dataset for now, but as the number of triples in the ontology grows, the performance difference may be much more pronounced. Stream reasoning queries could also have been used to activate decisions on the fly, without storing data in the ontology, however, the ontology supports combining the stream reasoning with other static data pre-captured in the ontology including the control actions to be recommended to the occupants.

In order to asses the overall effectiveness of the system, we compared the number of times that the system raised alarms for predicting pollution with the number of times that the corresponding records in the actual data specifies that both half hour situations and one hour situations are "*Poor*". Out of the 132 observations in the test data, the "*Poor*" condition is satisfied 52 times, however, the system raised alarms 59 times, giving 7 (11.86%) false alarms. The false alarms were found to be due to false positive predictions by the situation prediction component.

5.5 Summary

Although the concept of proactive computing is not new [6, 9], many SSW monitoring applications are still designed in reactive manners. The reason is perhaps due to the fact that the predictive methods, such as predictive reasoning [92, 93], that are native to Semantic Web technologies, are still emerging [87]. And although recent works in the stream reasoning community offer support for incorporation of heterogeneous data stream sources, more work is needed, especially on the approaches to incorporate predictive models within the processing space of a stream reasoning framework for SSW applications. This study proposes an architecture that attempts to fill this gap.

This chapter has presented an approach to achieve proactive monitoring and control in the SSW framework by incorporating a statistical prediction model in the processing space of a stream reasoning framework. The proactive monitoring and control approach was demonstrated with an indoor air quality use case and data streams from a real life use case in a low-cost residential setting. Secondly, a sliding window approach that employs MLP classifier for predicting indoor $PM_{2.5}$ pollution levels from low cost sensor observation data streams was presented.

The proposed framework provides a mechanism to combine the high accuracy and performance of statistical predictive techniques and the expressiveness of semantic analytic techniques for proactive monitoring and control. The architecture was shown to be effective for combining both stream reasoning processes and the outputs of predictive models for predicting situations of interest.

The decision processing mechanism of the proactive architecture is demonstrated in this chapter by reasoning on the ontology in Sections 5.3 and 5.4. The next chapter presents a study that investigates a decision processing mechanism, which incorporate the classical principles of probability and utility for the proactive SSW architecture.

Chapter 6

Incorporating MDP Theory in a Proactive SSW Framework

This chapter presents the incorporation of MDP theory into the proposed proactive SSW framework in an attempt to enhance consistency and coherence in decision processing. This chapter also presents the second use case in which the framework for proactive monitoring and control is evaluated. The use case is in the area of demand side management in a smart grid. Monitoring and control of demand load in order to avoid load shedding is the focus of the use case. A successful application of the framework to the problem of demand-side management, especially, dynamic load shedding is demonstrated using real world smart grid data.

Real time decision processing in order to take control of anticipated situations (*proactive control*) based on sensor data is a key challenge of a proactive application in the SSW [146]. Decision processing with Semantic Web technology, especially ontology driven applications, has been largely treated by evaluating semantic queries on the rules in the ontology to infer appropriate decisions. While decision rules support expressiveness in terms of the knowledge encoded in the rules, classical decision theory is built upon axioms of probability and utility. Probability theory supports the framework for coherent assignment of beliefs with uncertain information, and utility theory provides a set of principles for consistency in processing preferences and decisions [71]. In a dynamic environment, maintaining consistency and coherence in decision processing is a non trivial problem. The approach supports a tighter incorporation of components of the proposed proactive SSW framework.

The rest of this chapter is organized as follows. In Section 6.1, we present the framework. In Section 6.2 we introduce the application use case and highlight the need for a proactive monitoring and control system. Section 6.3 discusses the implementation of the system components and demonstrates how a MDP decision model is incorporated in the SSW framework. Section 6.4 presents the analysis and evaluation and in Section 6.5 we present a future direction. Section 6.6 summarizes the chapter.

6.1 The SSW framework for proactive monitoring and control

6.1.1 Abstract architecture

The three layered abstract architecture was introduced in Chapter 3. Figure 6.1 is the same as Figure 3.2 except that it highlights the components this chapter focuses on (see green border in 6.1). This study extends the control layer of the framework with a hybridized decision processing mechanism.

Figure 6.2 shows the technologies incorporated in the proposed proactive SSW framework for monitoring and control applications. The dotted line shows the focus of this chapter and the green circle shows the extension in this chapter.

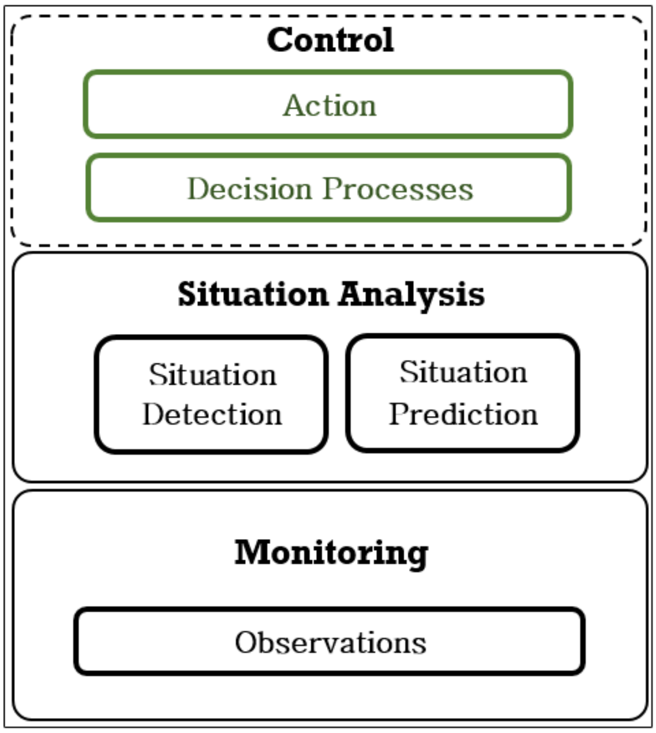


Figure 6.1: Abstract architecture for a proactive SSW framework.

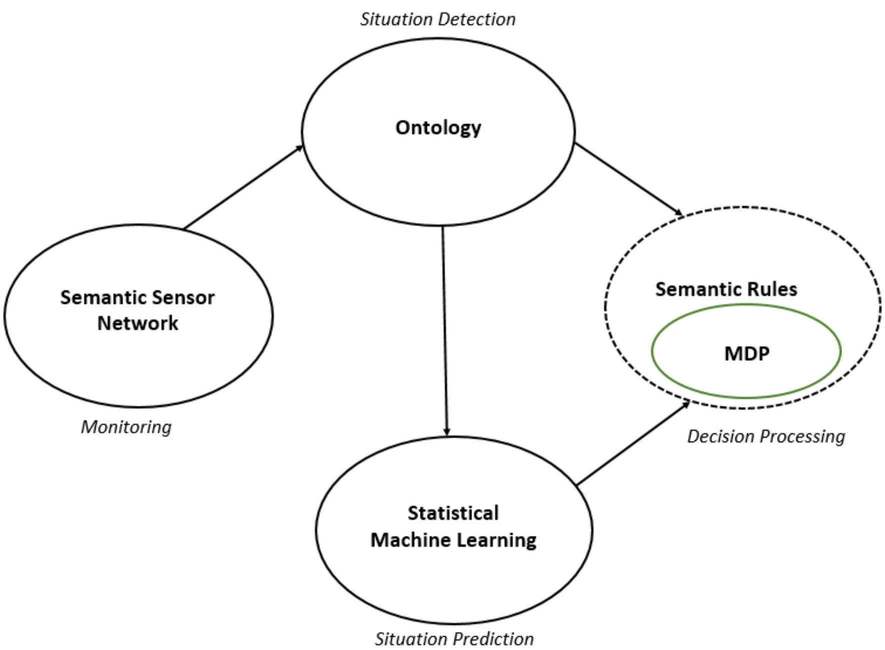


Figure 6.2: Technologies incorporated in the proactive SSW framework

The data flow among the main components of the framework is illustrated in a data flow diagram shown in Figure 6.3. The decision processes which is the focus of this study is highlighted in green.

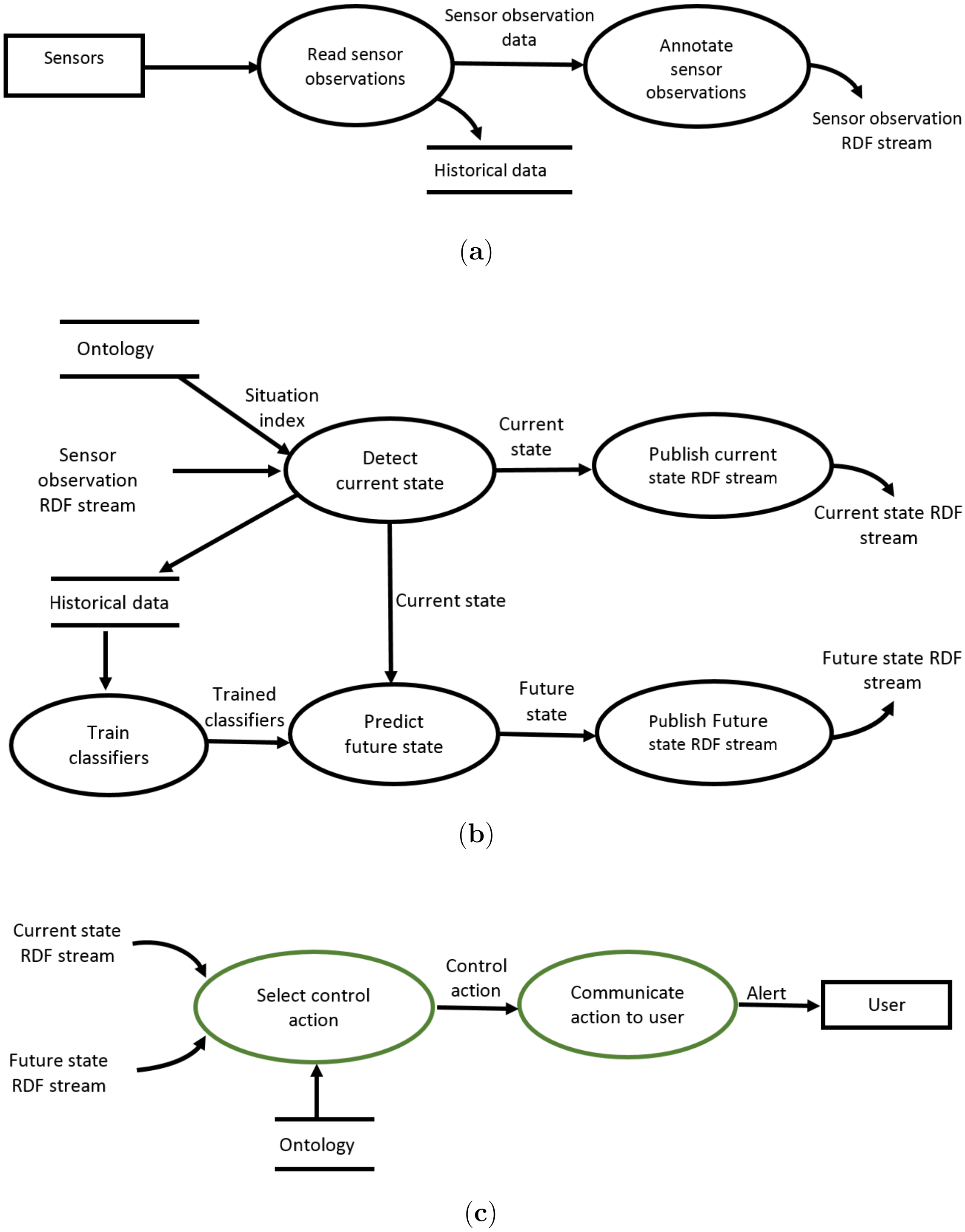


Figure 6.3: Level 1 Dataflow diagram of the proactive SSW framework: (**a**) Monitoring (**b**) Situation analysis (**c**) Control

The implementation details of the proactive decision processing is presented in Section 6.3. First, we present an overview of the proactive SSW framework and highlight the decision problem, to motivate the approach for the decision modeling and analysis.

6.2 Application use case

Power demand load management is a known problem in many countries, and it is currently receiving a lot of research attention [22, 24, 143]. *Load shedding* in a community involves cutting off the electric power supply in some part of the community for an arbitrary period of time when the demand overwhelms the power supply capacity in order to maintain stability and prevent total grid collapse. For example, in South Africa, Eskom, the main power utility company in the country, has noted a rapid increase of approximately 4000MW on the national grid during the winter seasons, due to the increased demand of electricity for space heating, geysers, pool pumps operations and cooking needs in residential buildings[1]. The peak period has been noticed to be in the evening between 5:00pm and 9:00pm when most occupants have returned home from daily engagements. However, in the summer seasons, high demand is usually experienced all through the day perhaps due to the use of air conditioners to cool down spaces. It is noted that avoiding the use of geysers, pool pumps, and cooking appliances during the peak periods can significantly reduce the demand on the grid. Hence, the goal of the utility company is to stimulate proactive actions among households in order to avert possible load shedding during the peak periods or at least minimize its occurrence.

During the high demand periods, a pre-scheduled time table is maintained for load shedding in municipalities across the country where load shedding is required. The time table groups houses into *blocks* and specifies when load shedding will be taking place in each of the blocks. A mobile application is also provided to inform the dwellers in the communities of the possible load shedding schedules. Also, Eskom publishes on its website the current state of power supply with steps on how individual houses can reduce power demand. Other efforts include information on the website stating the current situation of the demand load on the national and

[1] http://loadshedding.eskom.co.za/loadshedding/FAQ

provincial power grid and demand load forecasts for up to 24 hours. Information on the national television screens showing the current load status and messages on responsible usage of electricity in households is shown in Table 6.1.

Table 6.1: Messages in the existing system

Level	Load status	Messages
1	Stable	Commendation on efficient power uses
2	Limited	Switch off unnecessary lights to avoid strains
3	Strained	Switch off unnecessary lights, geysers, pool, pumps
4	Severely under pressure	Switch off unnecessary lights, geysers, pool pumps and all other appliances to prevent load shedding
5	Load shedding in some places	Switch off unnecessary lights, geysers, pool pumps and other appliances to prevent load shedding from spreading

In the context of this use case, a proactive SSW monitoring and control system is needed to continually monitor the demand load situation in the block of houses and avert pending load shedding when necessary. Smart meters as sensors can record power demand in a household and send same to the utility at specified time intervals for processing. The system can continuously monitor in real time, power demand from houses in the communities through the network of smart meters (*smart grid*). It can predict future power load situations and possible need for load shedding based on historical data and provide proactive warnings to the community to stimulate *proactive actions* as regards usage of appliances in order to avert load shedding and forestall its unwanted consequences. In this manner the system can stimulate more responsible usage of power in the communities and reduce the frequency of load shedding pro-actively.

6.2.1 System design

Four time periods each of 6 hours durations are defined, as *'morning'*, *'noon'*, *'evening'* and *'night'* in a 24 hours (see Table 6.2). The time periods allow tracking of load situations in each period.

Situation detection in the proactive SSW framework relies on an ontology driven index to translate quantitative sensor data to qualitative states. A *load index* is

Table 6.2: Time periods specifications for the proposed system

Period	**From**	**To**
'morning'	4:00 Hour	10:00 Hour
'noon'	10:00 Hour	16:00 Hour
'evening'	16:00 Hour	22:00 Hour
'night'	22:00 Hour	4:00 Hour

defined which classifies the state of power demand load in the monitored block of houses at any given time into one of 4 classes such as *'Stable'*, *'Strained'*, *'Under-Pressure'* and *'OffLimit'* (see Table 6.3). The specific values of the index depends on the power available to each of the blocks given the capacity of the supplier. The ontology driven index also serves to provide class labels for the dataset used to train classifiers for situation prediction.

Table 6.3: Load indices for situation detection

Hourly power consumption (KWh)	Situation
$<= 18$	*Stable*
> 18 and $<= 28$	*Strained*
> 28 and $<= 32$	*UnderPressure*
> 32	*OffLimit*

6.2.2 The hybridized decision processing model

The decision processing is required to select appropriate control actions the occupants need to take in order to prevent possible load shedding. The decision processing in this context is data-driven. The data that is driving the decision in the framework is the outputs of situation detection (*current state*) and situation prediction (*future state*). The hybridized decision processing is modeled with both semantic web rules and MDP planner.

Rule based reasoning with semantic rules: Reasoning on an ontology in the Semantic Web involves deriving logical consequences from facts and axioms asserted in the ontology. Semantic rules allow combining ontologies with sets of assertions in the ontology for rule based reasoning and analysis. Semantic rules are usually expressed with two main parts, the *antecedents* and the *consequents*, such that if the statements in the antecedent part are true, then all the statements in the consequent parts are executed.

The proposed proactive SSW framework imports and extends the SSN ontology [43]. In this study, the ontology is extended with concepts and relationships to support reasoning on the decision processing extension. A fragment of the ontology showing the data model which is based on the SSN ontology is shown in Figure 6.4.

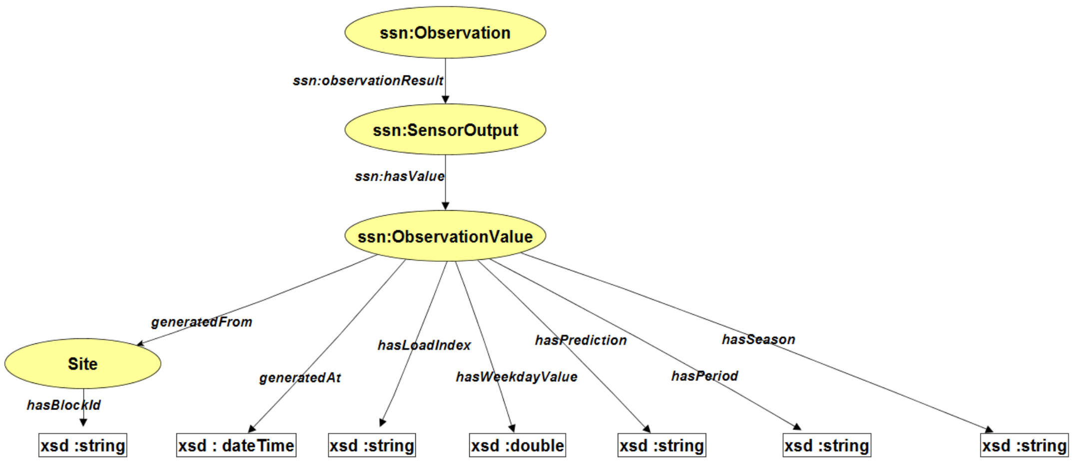


Figure 6.4: Fragment of the ontology showing data model based on the SSN ontology

MDP: MDPs have proven useful for modeling sequential decision making in stochastic environments [38]. A basic MDP model consists of: a finite set of states S which defines all the possible states of the world; a finite set of actions A which are the available actions in the world; a state transition function T that defines the relationship between actions and states; and a reward function R which gives a measure of preference or desirability of states and actions.

More formally [38], a MDP can be represented as follows.

S = set of states;

A = set of actions;

$T : S \times A \times S \to \mathbb{R} = Pr(s' \mid s, a)$;

$R : S \times A \to \mathbb{R}$.

In this study, the MDP model is used to decide on control actions to be implemented by occupants of the monitored block of houses in order to prevent load shedding with possible minimal discomfort to the occupants.

6.3 Incorporation of MDP theory in the proactive SSW framework

The application is implemented in the Java language in an Eclipse Integrated Development Environment, as an extension of the ontology driven testbed reported in Adeleke et al. [5]. The situation prediction component is implemented using the Wakaito Environment for Knowledge Analysis (WEKA) library. JENA and C-SPARQL libraries are also incorporated in the system for the ontology and stream reasoning components respectively. The MDP extension to the decision component runs as a Python module and is connected to the framework over a TCP/IP socket. The implementation details of the various system components are discussed subsequently.

6.3.1 Observation data

This work is evaluated with 3 years' (June, 2013 to June, 2016) hourly real live smart meter data obtained with permission from Pecan street dataport, a multi-institutional smart grid demonstration project in Austin, Texas, that has been widely adopted for smart grid research [111, 126]. The data consists of 35,111 data points. The smart meter readings of 12 different houses from the repository is aggregated to represent a monitored block of houses.

Figure 6.5 shows the visualization of the aggregated data. A typical variation of power consumption load across the 4 seasons of a year (2015) is shown in Figure 6.6.

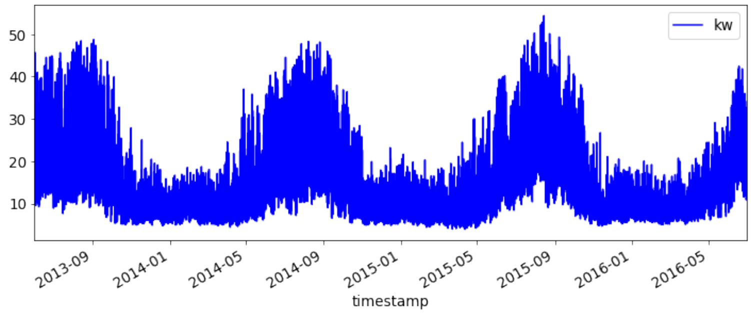


Figure 6.5: Aggregated hourly smart grid data of a block of 12 houses for 3 years.

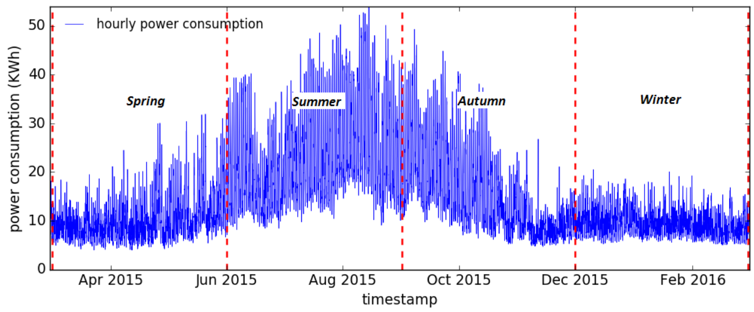


Figure 6.6: Typical variation of power load in different seasons of a year.

6.3.2 Situation detection

An ontology driven index (*load index*) which interprets the current demand load to a qualitative states (situation) is implemented as demonstrated in Chapter 4. The load index is queried each time it is necessary to detect the current situation. Figure 6.7 demonstrates how power consumption cuts across the load index in a typical year.

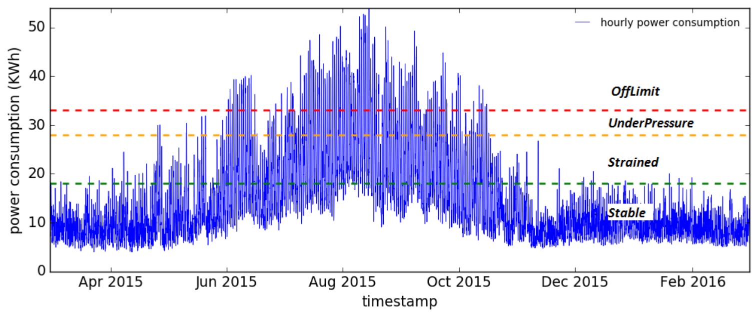


Figure 6.7: Power consumption across the load indices in a typical year.

6.3.3 Situation prediction

Employing machine learning techniques as presented in the previous chapter, a classifier is then made to classify the situation for the next time step in the future, into one of three non overlapping classes such as, *Stable*, *Strained* and *UnderPressure*. The classes are as described for the load index, except that both the *UnderPressure* and *OffLimit* classes are merged as *UnderPressure* for the prediction task.

J48 (see Section 5.2.2), a decision tree classifier was found suitable and used to implement the situation prediction in this work [122]. The features for building the classifiers include month, day, hour, weekday, season, current demand load value, the demand load value for the same hour the previous day and the demand load value for the same hour two days before. The latter is important because demand load prediction task is known to be influenced by the value of load at the same time of the previous day [7].

The performance of the classifiers was evaluated with the popularly used metrics for multi-class classification tasks such as *Accuracy,* $Precision_M$*,* $Recall_M$ and $F\text{-}score_M$ [120, 138] (see Section 5.2.3). For a multiclass classification task, these are calculated as a macro averaging of the metrics for the individual classes [138]. A $Precision_M$ of 0.862 and $Recall_M$ of 0.860 were achieved in the classification task.

6.3.4 Decision processing and action

The goal of the decision processing mechanism is to avert possible progression towards load shedding. That is, to select and communicate appropriate control actions to the occupants in order to avert any predicted unwanted load situations. The control actions are modeled as restrictions on the use of certain appliances to be imposed on the occupants, which in turn will influence the load situation.

The state of the world defined for the MDP model is a quadruple $\langle s, d, p, i \rangle$, where $s \in \{autumn, spring, winter, summer\}$ is the present season in the year and $d \in$

$\{0,1\}$ specifies whether it is a weekday (1) or a weekend (0). $p \in \{morning, noon, evening, night\}$ is the period of the day, while $i \in \{Stable, Strained, UnderPressure, OffLimit\}$ is the load index that indicates the demand load situation on the grid.

The available control actions are : *NoAction*, which imposes no restriction to the occupants; *SwitchOffUnusedLightNApps*, a low restriction control action compelling the occupants to switch off all unused light and appliances; *SwitchOffGeysers*, a moderate control action that forbids the use of geysers; *SwitchOffPoolPumps* which is also a moderate restriction control action that forbids the use of pool pumps; *SwitchOffCookingNHeating*, which is also a moderate restriction control action that requires the occupants to switch off all cooking and heating appliances; and finally, *SwitchOffAllApps* is a high restriction control action that forbids the use of all appliances except light. Further details on the functionality of the decision processing is explained the next section. Figure 6.8 illustrates the end to end implementation of the framework.

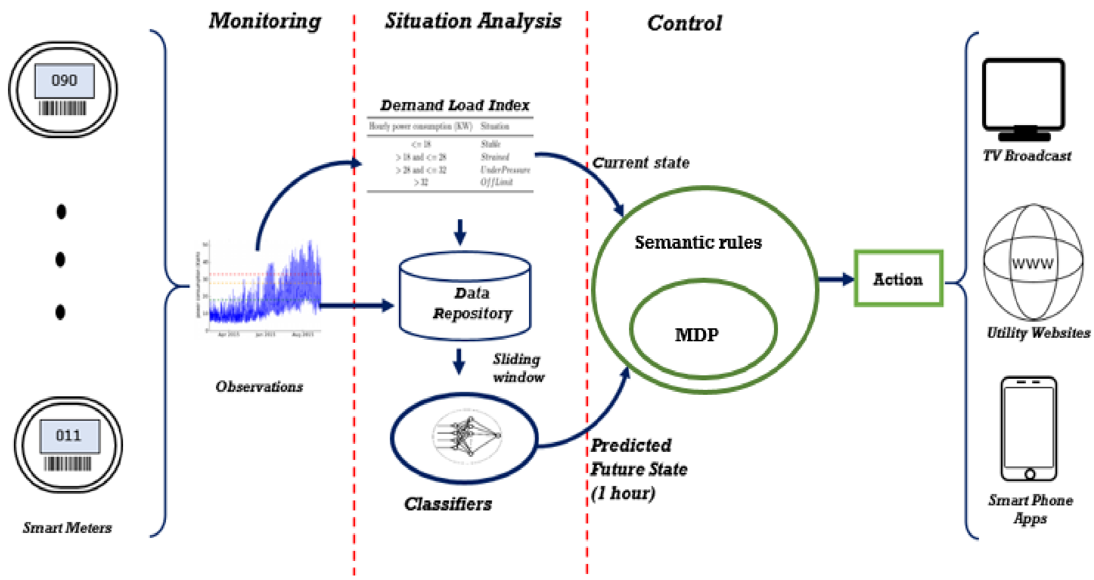


Figure 6.8: Incorporation of the decision processing in the SSW framework.

6.4 Analysis and evaluation.

In order to evaluate the effectiveness of the decision mechanism and the added value of incorporating the MDP planner with an ontology driven system, experiments were performed. In the experiments, the use case scenario was modeled to simulate the possible load situations in the block of houses, and the actions of the decision mechanism. In the experiments, it is assumed that suggested actions are carried out. First, we compare the semantic rules and MDP approaches and then presents the hybrid approach.

The experiment is performed on the hourly smart grid data covering the months of June to November for the year 2014 and 2015. These are the months of the year when demand load gets to the peak (see Figure 6.6). The performance of the decision mechanism is evaluated by two criteria. First, the ability to avert load shedding and second *occupant satisfaction* OS. To determine the latter, a penalty P is attached to each of the control actions which vary according to the level of dissatisfaction it causes the occupants, and a reward R is given based on the resultant state of the load index after performing suggested action (see Table 6.6). Occupant satisfaction is then calculated by subtracting P from R in each case, that is, $OS = R$ - P. Table 6.5 shows the costs and rewards scores used in the experiments. Average daily OS is calculated and the mean OS values for 5 runs is estimated and plotted against *Datetime* for analysis. We first compare the performance of the semantic rules and the MDP separately and then compare that with the hybrid mechanism.

6.4.1 Comparison of semantic rules and MDP decision processing

Semantic rules: Seven semantic rules were implemented with Jena rule engine to specify the best actions to perform in each possible situations, the rules are based on domain expert knowledge. The logic of the rules are shown in the listings below.

```
Rule 1. hasLoadIndex(?situation, 'stable')
-> controlAction(?situation, ?NoAction)

Rule 2. hasLoadIndex(?situation, 'Strained'),
-> controlAction(?situation, ?SwitchOffUnusedLightNApps)

Rule 3. hasLoadIndex(?situation, 'Underpressure'),
hasPeriod(?situation, 'morning'),
-> controlAction(?situation, ?SwitchOffPoolPumps)

Rule 4. hasLoadIndex(?situation, 'Underpressure'),
hasPeriod(?situation, 'noon'),
-> controlAction(?situation, ?SwitchOffGeysers )

Rule 5. hasLoadIndex(?situation, 'Underpressure'),
hasPeriod(?situation, 'evening'),
-> controlAction(?situation, ?SwitchOffCookingNHeating )

Rule 6. hasLoadIndex(?situation, 'Underpressure'),
hasPeriod(?situation, 'night'),
-> controlAction(?situation, ?SwitchOffCookingNHeating )

Rule 7. hasLoadIndex(?situation, 'Offlimit')
-> controlAction(?situation, ?SwitchOffAllapps)
```

MDP: The MDP planner takes in the current state of the world as the initial state. The probability distribution shown in Table 6.4 was made into a transition function. The process is made to plan for 3 horizons of 20 minutes each. The reward function is based on the costs and rewards towards occupants satisfaction. After performing the suggested action, the process observes the post-action load index which represents the effectiveness of the action. In order to estimate *occupant satisfaction OS*, the action performed and the post-action load index are scored according to Table 6.5

and Table 6.6 respectfully. These two tables define the MDP reward function.

Table 6.4: Probability distributions modeling effect of control actions on load indices and used as transition function.

Action	Current load index	Probabilities of load index after action			
		Stable	*Strained*	*UnderPressure*	*OffLimit*
NoAction	***Stable***	0.9	0.1	0.0	0.0
	Strained	0.2	0.3	0.3	0.2
	UnderPressure	0.0	0.2	0.6	0.2
	OffLimit	0.0	0.0	0.1	0.9
SwitchOffUnusedLightNApps	***Stable***	0.8	0.2	0.0	0.0
	Strained	0.7	0.2	0.1	0.0
	UnderPressure	0.1	0.5	0.4	0.0
	OffLimit	0.0	0.1	0.5	0.4
SwitchOffGeysers ***SwitchOffPoolPumps*** ***SwitchOffCookingNHeating***	***Stable***	1.0	0.0	0.0	0.0
	Strained	0.7	0.2	0.1	0.0
	UnderPressure	0.6	0.3	0.1	0.0
	OffLimit	0.3	0.4	0.3	0.0
SwitchOffAllApps	***Stable***	1.0	0.0	0.0	0.0
	Strained	1.0	0.0	0.0	0.0
	UnderPressure	1.0	0.0	0.0	0.0
	OffLimit	1.0	0.0	0.0	0.0

Table 6.5: Penalties of actions for estimation of occupant satisfaction

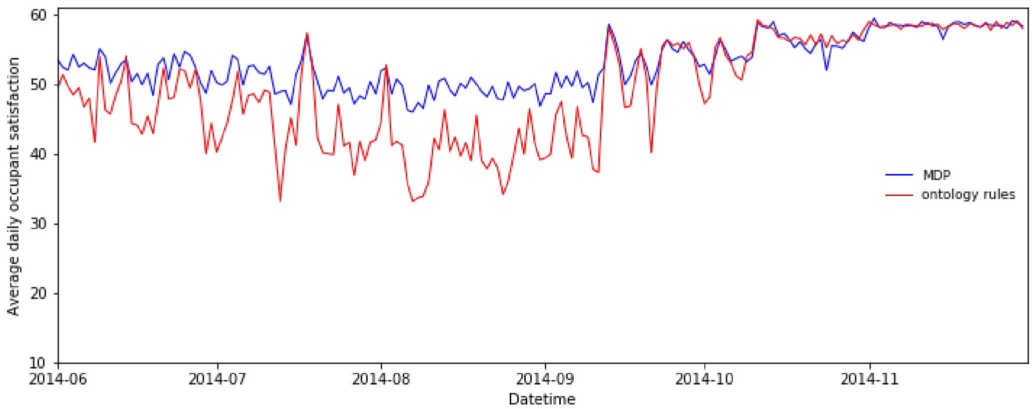
Action	P			
	Morning	**Noon**	**Evening**	**Night**
NoAction	0	0	0	0
SwitchOffUnusedLightNApps	5	5	5	10
SwitchOffGeysers	15	10	10	5
SwitchOffPoolPumps	5	5	5	5
SwitchOffCookingandHeating	10	10	10	5
SwitchOffAllApps	55	55	55	55

Table 6.6: Rewards of post-action load index for estimation of occupant satisfaction

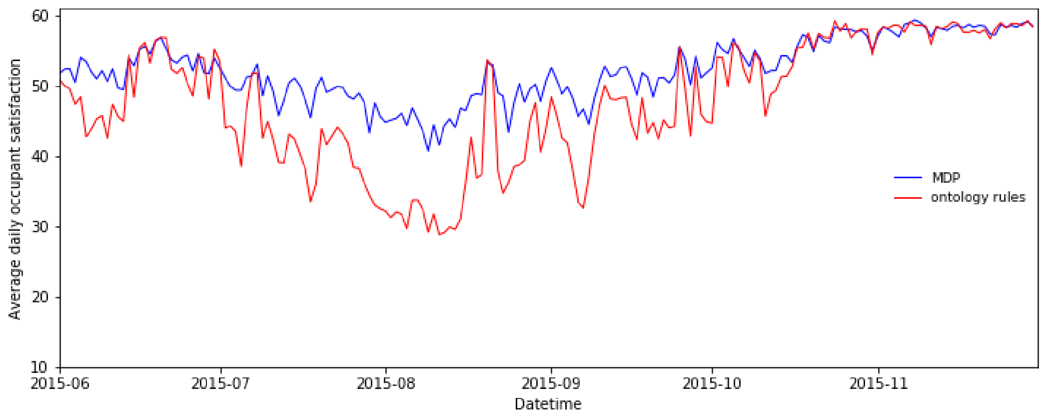
post	R
Stable	60
Strained	45
UnderPressure	15
OffLimit	0

The experiment is performed on the smart grid data covering the months of June to November of 2 years (2014, 2015) and average daily occupant satisfaction is calculated in each case. The mean values of 5 runs is estimated for analysis for both the semantic rule and MDP mechanism.

Result: Both the MDP and the semantic rules decision mechanisms are able to suggest control actions that averted the need for load shedding based on the defined set of actions. However, the MDP process coherently gave a more consistent and higher occupant satisfaction. This is more so during the high peak periods. Figure 6.9 shows the average daily occupant satisfaction for the 5 months of the 2 years (see Figure 6.9).



(**a**)



(**b**)

Figure 6.9: Line charts showing occupant satisfaction in the monitored block of houses for both the MDP and the semantic rules mechanisms (**a**) June to November 2014 (**b**) June to November 2015.

6.4.2 The hybrid decision mechanism

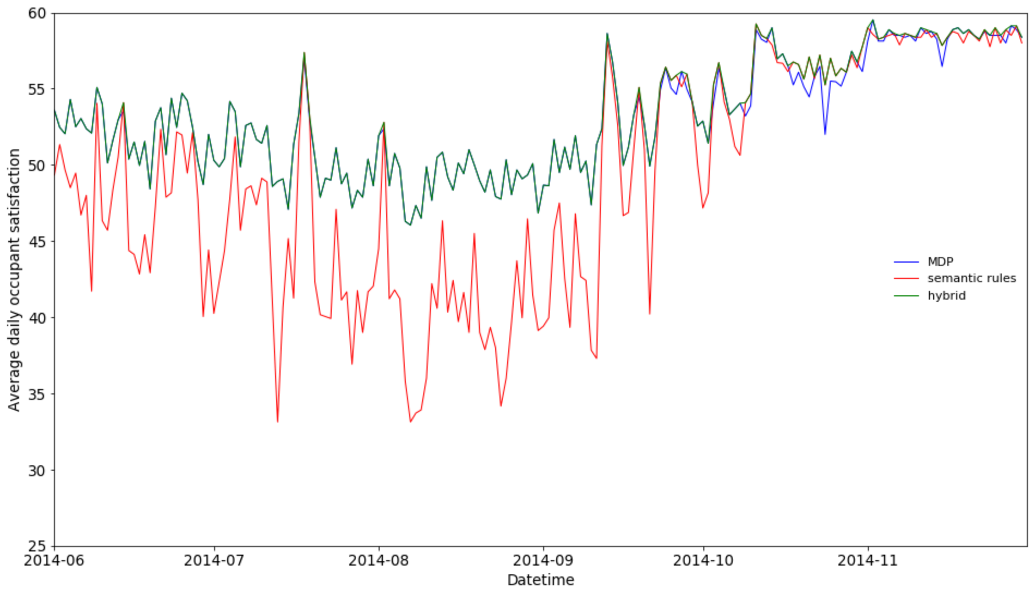
The semantic rules in the previous experiment were further enriched to include predicted states, and the rule is made to call the MDP mechanism when planning is necessary. The mechanism takes in, both the predicted and the current state of the world. Semantic rules are defined to represent expert knowledge which specifies the course of action depending on values of the current state and the predicted state. When both the current state and the predicted state are *Stable*, the mechanism recommends *NoAction* and planning is not necessary. When either the current or the predicted state is *Strained*, the mechanism recommends *SwitchOffUnusedLightandApps*. The rule engine runs the MDP planner anytime either the current state or the predicted state becomes *UnderPressure* and the control action that the planner outputs is recommended. The logic of the semantic rules is shown in the following listing.

Semantic rules:

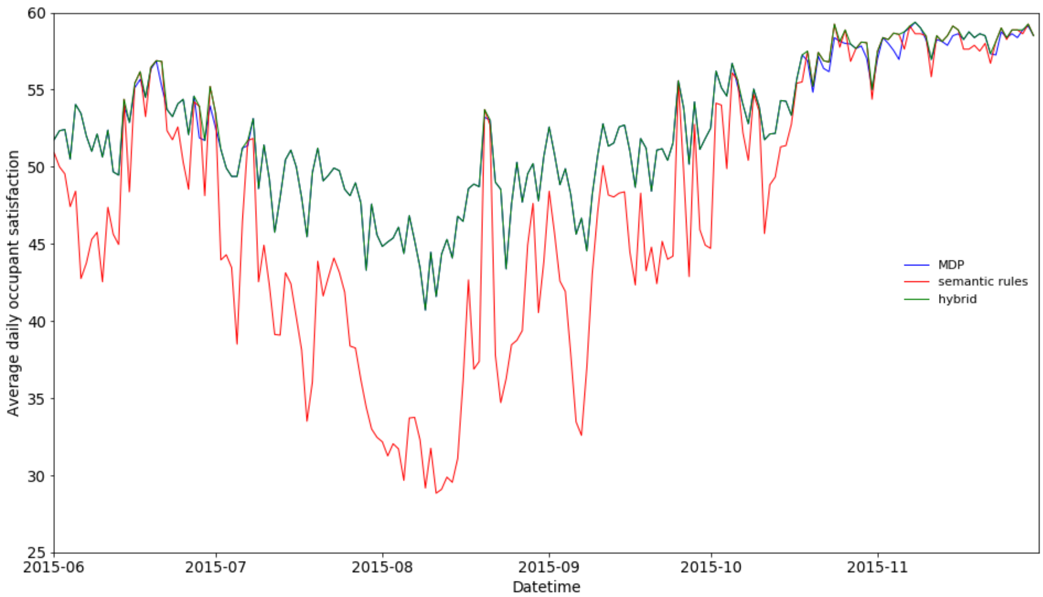
```
Rule 1. hasLoadIndex(?situation, 'Stable')
hasPredictedState(?situation, 'Stable"),
-> controlAction(?situation, ?NoAction)

Rule 2. hasLoadIndex(?situation, 'Stable')
hasPredictedState(?situation, 'Strained'),
-> controlAction(?situation, ?SwitchOffUnusedLightNApps)

Rule 3. hasLoadIndex(?situation, 'Stable')
hasPredictedState(?situation, 'UnderPressure'),
-> controlAction(?situation, ?MDP-planner)

Rule 4. hasLoadIndex(?situation, 'Strained'),
hasPredictedState(?situation, 'Stable'),
-> controlAction(?situation, ?SwitchOffUnusedLightNApps)

Rule 5. hasLoadIndex(?situation, 'Strained'),
hasPredictedState(?situation, 'Strained'),
-> controlAction(?situation, ?SwitchOffUnusedLightNApps)

Rule 6. hasLoadIndex(?situation, 'Strained'),
hasPredictedState(?situation, 'UnderPressure'),
-> controlAction(?situation, ?MDP-planner)

Rule 7. hasLoadIndex(?situation, 'UnderPressure'),
```

```
hasPredictedState(?situation, 'Stable'),
-> controlAction(?situation, ?MDP-planner)

Rule 8. hasLoadIndex(?situation, 'UnderPressure'),
hasPredictedState(?situation, 'Strained'),
-> controlAction(?situation, ?MDP-planner)

Rule 9. hasLoadIndex(?situation, 'UnderPressure'),
hasPredictedState(?situation, 'UnderPressure'),
-> controlAction(?situation, ?MDP-planner)

Rule 10. hasLoadIndex(?situation, 'Offlimit')
-> controlAction(?situation, ?SwitchOffAllapps)
```

Result: Figure 6.10 shows the average daily occupant satisfaction of the hybrid mechanism for the 5 months of the 2 years. From the figure, the hybrid mechanisms optimizes the best performance over the two mechanisms individually. The rules also serves as a means of incorporating the MDP mechanism to the framework.



(**a**)



(**b**)

Figure 6.10: Line charts showing occupant satisfaction in the monitored block of houses for the MDP, semantic rules and the hybrid mechanisms (**a**) June to November 2014 (**b**) June to November 2015.

6.5 Possible further incorporation of the MDP theory into the framework

The machine learning based situation prediction component gives prediction of possible future states. The prediction classifies the next hour state into one of three classes (*Stable, Strained* and *UnderPressure*). The classification can be expressed as a probability distribution over the three classes, given the current state, in which the class that has the highest probability is predicted. Thus, the probability Pr of predicted state s' given the current state s can be expressed as $Pr(s' \mid s)$. Pr here represents the probability that s' will occur in the next time step if the system does not implement any control action. Hence, the probability of the predicted state can be expressed in terms of control-actions $a \in A$ as

$$Pr(s' \mid s, a = \mathit{NoAction}) = Pr(s' \mid s) \tag{6.1}$$

where *NoAction* implies that no control action has been implemented. This can be generalized as

$$Pr(s' \mid s) = Pr(s' \mid s, a). \tag{6.2}$$

Equation (6.2) shows that $Pr(s' \mid s)$ which is a product of machine learning component relates the current state and control action to the future state in the same manner as the transition function of an MDP. Hence, it can be incorporated as a transition function in an MDP process for a dynamic control decision processing. Although the probability distribution provided by the classifier is valid for only when no action has been taken, the distribution can be scaled by some functions over different magnitudes of control actions. A change in a will influence the occurrence of s' in some manner. This can be exploited for selecting appropriate control action among the available ones in order to control the occurrence of the future state in favor of the desired ones.

Recall that MDP transition function T can be expressed as,

$$T(s, a, s') = Pr(s' \mid a, s) \tag{6.3}$$

where $Pr(s' \mid a, s)$ is the probability with which s' is reached via action a.

a can be rewritten as a function Act of s, s' pair

$$Act(s', s) = a. \tag{6.4}$$

Equation 6.3 then becomes

$$T(s, Act(s', s), s') = Pr(s' \mid a, s). \tag{6.5}$$

For a MDP transition function a valid probability distribution is required over all possible $s' \in S$ and $a \in A$, that is,

$$\sum_{s' \in S} Pr(s' \mid a, s) = 1. \tag{6.6}$$

Note that the $Pr(s' \mid a, s)$ given by the machine learning classifier is only valid for when *NoAction*. Hence when the prediction from the classifier coincides with the desired state, the probability distribution given by the classifier can be used to suggest a *NoAction* action. However, when the prediction from the classifier does not coincide with the desirable state, the probability distribution from the classifier is not valid to select the appropriate action.

In order to overcome this problem a *scaling function* can be introduced into the Equation 6.5 such that h composes Act. When the s' coincides with the desired state, there is no need for a control-action and no need for scaling. Whenever the predicted state is not the preferred state, h scales the distribution over different magnitudes of a needed to achieve the occurrence of the desired state given s and s'

while adhering to the constraint expressed by (6.6).

An obvious limitation of this approach is that the transition function will only be valid within the percentage of accuracy of the machine learning classifier that is providing the probability distribution. More work is still needed to validate the usefulness of this approach and evaluate it's efficacy. Hence, it is left for future work.

6.6 Summary

This chapter has presented an approach to incorporate MDP theory into the proactive SSW framework. The approach incorporates the planning capabilities of the MDPs in domain expert knowledge for coherent and consistent proactive decision making.

Incorporating the notion of probability and utility, the axioms on which classical decision theory is based, in a proactive SSW framework is a step forward in developing SSW applications. A prototype application of the proactive SSW framework to address the problem of demand-side management, especially load shedding prevention has been demonstrated, with real live smart grid data. This research suggest that the prototype application of the proposed proactive SSW framework can avert load shedding and also improve consumer satisfaction on the smart grid, within the context of SSW.

Chapter 7

Discussion and Conclusion

This thesis proposes a SSW framework for proactive environmental monitoring and control. Designing and developing proactive monitoring and control applications requires integrating and incorporating different techniques for supporting situation detection, situation prediction, decision making and planning. The proposed SSW framework incorporates ontologies to facilitate situation detection from streaming sensor observations, statistical machine learning for situation prediction and MDP for decision making and planning in a SSW framework. The efficacy of the proposed framework was evaluated through the design and development of two different prototype applications.

Although the idea of SSW is not new, most current SSW frameworks lack two essential mechanisms required to achieve proactive control, namely, mechanisms for anticipating the future and mechanisms for coherent and consistent decision processing and planning. Furthermore, application frameworks that provide tools and techniques for rapid application development is a challenge in the SSW community [44]. This research fills these gaps. The framework proposed in this thesis can support development of proactive SSW applications for environmental monitoring and control.

7.1 Summary of results

The main contribution of this research is a new SSW framework for developing proactive environmental applications, that is, applications that avert unwanted environmental situation before they occur. The framework allows combining the performance of advanced statistical machine learning techniques and the expressive analytic techniques of SSW technologies with the coherent and consistent planning capability of MDP theory to analyze streaming sensor observations. The framework and its key components are evaluated through the development of two different prototype applications.

7.1.1 Use case 1: Indoor air quality

The first use case which was in the area of indoor air quality was carried out along with an occupational health research group. Monitoring and control of unhealthy indoor air quality was made the focus of the use case. A prototype application of the framework was developed and evaluated with real life data. This use case was carried out in two phases.

Monitoring and control of PM_{10} was first used to evaluate the development of a SSW monitoring application, a prototype of the proposed framework that does not include situation prediction. The system was able to monitor and detect PM_{10} pollution situations, and when an unhealthy situation is detected, the system was able to provide control actions to abate the unhealthy situation. This phase of the use case evaluates the development and use of ontologies for environmental monitoring and control.

The three focus areas that was successfully evaluated in this phase of the use case include: adequate representation of sensor observations in the ontology; support for reasoning services for analyzing sensor data to detect target situations through the use of an ontology driven index; and determining control actions to mitigate unwanted situations through reasoning on the ontology. The details of this use case is described in Chapter 4.

The framework at this stage was a reactive one, it demonstrated the use of ontologies for a monitoring and control application.

Proactive monitoring and control of $PM_{2.5}$ was used to evaluate the incorporation of the machine learning model for situation prediction. The prototype application here was an extension of the application in the first phase of this use case. The evaluation in this phase was focused on two main areas. The first was the exploration of statistical machine learning for situation prediction. In the use case that was reported in Chapter 5, a sliding window technique for predicting the future from streaming sensor data was proposed. The situation prediction approach demonstrated how to structure sensor data for the prediction tasks and for on-line retraining of the classifiers; how to select an appropriate classifier for the prediction task and how to select appropriate sliding window sizes for situation prediction. The second area evaluated was the incorporation of situation prediction in the SSW framework. This was achieved using C-SPARQL a stream reasoning engine. The use of stream reasoning technique ensures that only target situations are recorded in the ontology, which is in turn important for better query performance on the ontology.

Incorporating the statistical machine learning model in the framework enabled it to predict future situations and process control actions before their occurrence, thereby making it a proactive framework. In the application use case, the system could then monitor short term future $PM_{2.5}$ pollution situations and provide control actions before the situations occur.

Evaluation results show that although the MLP classifiers demonstrated slightly better performance than BN in predicting $PM_{2.5}$ pollution situations, it is not as scalable in terms of model update time as the BN when the data set size increases.

The application prototype developed in the indoor air quality use case was done as a pilot project with an occupational health research group in the university (see Chapter 5). Hence, the framework can be used by occupational health researchers and practitioners as an early-warning system to prevent excessive exposure of occupants to indoor pollutants.

7.1.2 Use case 2: Demand side management

The second use case was in the area of demand side management on the smart grid. A prototype application for monitoring and control of electricity usage in blocks of residential houses was developed. The goal of the application was to prevent strain on the national grid thereby preventing possible power cuts. The focus of this use case was to evaluate coherence and consistency in decision making and planning.

The proposed framework promotes the use of an ontology driven index for situation detection. The implementation of an ontology driven index for detecting demand load situations in this use case is the second successful evaluation of this approach. The index was able to detect the qualitative power demand states from quantitative sensor (smart meters) data. The novel technique that allows dynamic labeling of the dataset using the ontology driven index for retraining the machine learning model was also evaluated.

This application further evaluates the sliding window technique for situation prediction that was introduced in Chapter 5. The performance of this approach using J48, a decision tree classifier to predict the next hour demand load on the smart grid suggests that the sliding window approach can be used to predict power demand load situations.

The use case demonstrated an approach to incorporate an MDP planner in a SSW in order to enhance rule based reasoning with coherence and consistency in decision processing. The approach demonstrated a MDP-semantic rules hybrid mechanism which is described in more detail in Chapter 6.

Evaluation results shows that the prototype application can effectively avert load shedding and increase occupant's satisfaction. The prototype application developed and presented in Chapter 6, can be used by an utility manager for effective management of power demand in blocks of residential houses to prevent strain on the smart grid.

7.2 Usage of the framework

The framework can guide a developer in designing and developing SSW applications for proactive environmental monitoring and control. It provides the required functional components in the different layers. Since the framework has three layers, namely, monitoring, situation analysis and control, we recommend application development to follow this order.

- **Monitoring layer:** The main component of this layer is observation. Relevant observations from the sensors monitoring the specific feature of interest in the environment should be identified.

  The framework reused and extended the SSN ontology as shown in Chapter 4. The ontology engineering steps provided can be followed to model concepts and relationships in an application domain to support the framework's components and reuse the SSN ontology.

- **Situation analysis layer:** There are two main components to be focused in this layer, namely situation detection and situation prediction.

  Situation detection. The proposed framework uses an ontology driven index to achieve situation detection. The steps provided in the two use cases can guide a developer to implement an ontology driven index for the specific situations of interest in an application domain.

  Situation prediction. The sliding window approach for situation prediction introduced in Chapter 5 provides detail steps that can be followed to model and evaluate and incorporate situation prediction, based on statistical machine learning, in the application domain.

- **Control layer:** This research proposes a hybrid MDP-Semantic rule decision processing mechanism for making coherent proactive decisions. The steps provided in Chapter 6 can guide a developer to develop a coherent and consistent decision mechanism.

  By following the steps above and using the proposed mechanisms a developer can design and implement an application for proactive environmental monitoring and control.

7.3 Framework analysis

7.3.1 Comparison with previous work

This section compares the framework proposed in this thesis with existing architectures and frameworks.

SSW architectures and frameworks typically do have several layers of semantic infrastructures and services [42, 44]. The SSW framework proposed in this thesis follows the generic SSN architecture presented by Compton et al. [42], which also has three layers, namely, sensor and data layer, processing layer and application layer. The sensor layer and the processing layer are similar to the monitoring and situation analysis layers respectively. Except that the processing included in our situation analysis layer has been streamlined to the focus of the layer, namely situation detection and situation prediction. However, while their architecture's third layer is the application layer, the third layer in our architecture is control.

The architecture proposed by Gray et al. [66], (see Section 2.3.3) a multi-tier service oriented architecture is similar to our architecture in that it was focused towards providing early warnings in the SSW. However, the architecture was focused towards orchestration of services. Although it leverages forecast services to anticipate the future through service integration, and provide early warnings, the architecture is different from our's, in that it does not include mechanisms to predict the future as a component of the architecture.

The SWAP architecture (see Section 2.3.3) is the most similar to the architecture proposed in this work. First, it's a three layer abstract architecture. Second, it can be used for developing SSW applications. However, SWAP is different from our proposed framework. Firstly, SWAP is an ontology-driven multi-agent system while our proposed framework is not focused towards developing agent based application. Secondly, although SWAP has incorporated machine learning classification in an application for classifying and detecting informal settlement, the application is a reactive one. It has not been focused towards predicting the future and acting in favor of the user. Hence, the approach proposed in this thesis for incorporating statistical machine learning in a SSW for proactive monitoring and control can be used with SWAP to develop a SSW application for proactive monitoring and control.

The framework proposed in this thesis is similar to that of Engel et al. [57]. Although their architecture also follows the proactive computing paradigm, it does not use SSW techniques. While our architecture is promoting tight incorporation between the components, their's is promoting decoupling of the components. They proposed use of an MDP for decision processing but not a hybrid one like ours.

The approach to achieve proactive control in this work is similar to that of Anaya [9] who sought to achieve proactive adaptation by incorporating predictive machine learning models in a self adaptive system. The author's approach is different by using fuzzy logic for decision making and the work does not use SSW technologies.

7.3.2 Interaction between the framework components

The proposed framework incorporates three key technologies for analyzing streaming sensor observations, namely, ontologies for situation detection, statistical machine learning for situation prediction and MDP for decision making and planning. This research demonstrates how these three components can be incorporated into a coherent architecture such that they enhance the functionality of each other.

The ontology-driven situation detection component provides supports for the functionality of the statistical machine learning components. First, it dynamically provides the current states which is an important attribute used by the machine learning classifiers to predict the future states. Second, it provides class labels for dynamic labeling of datasets used for on-line retraining of the classifiers. The latter is a novel technique that enhances system autonomy, and coherence in the outputs of the framework.

The hybrid decision processing component depends on the outputs of both ontology-driven situation detection and machine learning predictions for decision processing and planning. The combination of MDP and semantic rules for decision making is a novel technique that enhances rule-based reasoning on the ontology with coherent planning capability. The hybrid decision processing component further allows the machine learning predictions to inform planning and action selection by the MDP.

7.4 Review of objectives

Objective (i). The first objective of this thesis was to design and implement an ontology-driven SSW framework for monitoring and control applications. In fulfilling this objective, we investigated the use of ontology for monitoring and control (presented in Chapter 4). The ontology-driven framework also serves as a testbed for the incorporation and evaluation of other components of the framework.

Objective (ii). The second objective of this thesis was to design an approach to incorporate situation prediction based on statistical machine learning into a SSW framework. Anticipating the future is a core component of proactive control. To fulfill the objective, Chapter 5 presented an approach that leverage stream reasoning techniques to incorporate situation prediction into the SSW framework (see Chapter 5). The approach demonstrated how to structure situation prediction modeling with statistical machine learning techniques and how to incorporate the predicted situation in the SSW framework.

Objective (iii). The third objective of this work was to design and incorporate a coherent and consistent decision processing and planning mechanism in the proactive SSW framework. This is to introduce the notion of probability and utility into processing proactive decisions for coherence and consistency in the decision output. Chapter 6 has presented the study that investigates how to incorporate MDP theory in a proactive SSW framework. The study proposed a hybrid MDP-Semantic rule decision mechanism for making consistent proactive decisions and plans.

Objective (iv). Finally, the last objective of this study was to evaluate the proposed proactive SSW framework with real world environmental application use cases. In order to fulfill this objective, the design, development and evaluation of prototype applications of the framework in two different use cases has been presented in Chapter 5 and Chapter 6. The first application was for proactive monitoring and control of indoor air quality to avoid poor air quality situations. The second was for proactive monitoring and control of electricity usage in blocks of residential houses to prevent strain on the national grid.

7.5 Limitations and future work

7.5.1 Application use cases

The framework proposed in this thesis has potentials to facilitate proactive SSW applications development for different proactive environmental monitoring and control domains, within the context of SSW. It has only been evaluated through two environmental use cases, namely, indoor air quality and demand side management. Both use cases involved proactive monitoring and control of features of interests in residential home environments. More work is still required to evaluate its generality of use for other application domains.

7.5.2 Situation detection techniques

In this work ontologies are used for situation analysis and detection in the current time frame. Other techniques such as fuzzy logic and machine learning may also be used for situation analysis and detection. In future work it will be interesting to perform a more in-depth comparative study of the merits of using ontologies against these techniques.

7.5.3 Statistical machine learning prediction errors

The framework proposed in this thesis employs statistical machine learning to anticipate future states. Hence, the efficiency of the system can be impaired if the machine learning classifiers performs poorly. This can result in false alarms as demonstrated in Chapter 5. Possible future research could entail investigating how ontologies can be used to understand the pattern of errors in the machine learning predictions.

7.5.4 Incorporation of components

As mentioned in Section 7.3.2, an important added value of incorporating several technologies in the proposed SSW framework is the use of one component to enhance the functionality of the other. As such ontologies support the machine learning component, e.g. labeling historical data which can then be used for model construction. More work is required in this aspect. The transition function used by the MDP in the use case which was reported in Chapter 6 was modeled according to expert domain knowledge. A future direction could be to make the machine learning components also enhance the functionality of the MDP by providing probability distributions dynamically (see Section 6.5). This can improve the system dynamism and make the framework more coherent by learning the transition function from the streaming data rather than expert knowledge.

Bibliography

1. Abdalla, A., Hu, Y., Carral, D., Li, N., and Janowicz, K. An ontology design pattern for activity reasoning. In *Proceedings of the 5th International Conference on Ontology and Semantic Web Patterns-Volume 1302*, pages 78–81. CEUR-WS. org, 2014.

2. Aberer, K., Hauswirth, M., and Salehi, A. A middleware for fast and flexible sensor network deployment. In *Proceedings of the 32nd international conference on Very large data bases*, pages 1199–1202. VLDB Endowment, 2006.

3. Abowd, G. D. and Mynatt, E. D. Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 7(1):29–58, 2000.

4. Adeleke, J. A. and Moodley, D. An ontology for proactive indoor environmental quality monitoring and control. In *Proceedings of the 2015 Annual Research Conference on South African Institute of Computer Scientists and Information Technologists*, page 2. ACM, 2015.

5. Adeleke, J. A., Moodley, D., Rens, G., and Adewumi, A. O. Integrating statistical machine learning in a semantic sensor web for proactive monitoring and control. *Sensors*, 17(4):807, 2017. ISSN 1424-8220. doi: 10.3390/s17040807. URL http://www.mdpi.com/1424-8220/17/4/807.

6. Akbar, A., Carrez, F., Moessner, K., and Zoha, A. Predicting complex events for pro-active iot applications. In *Internet of Things (WF-IoT), 2015 IEEE 2nd World Forum on*, pages 327–332. IEEE, 2015.

7. Al-Shareef, A., Mohamed, E., and Al-Judaibi, E. One hour ahead load forecasting using artificial neural network for the western area of saudi arabia. *International Journal of Electrical Systems Science and Engineering*, 1(1):35–40, 2008.

8. Alcock, R., Manolopoulos, Y., et al. Time-series similarity queries employing a feature-based approach. In *In 7 th Hellenic Conference on Informatics, Ioannina*. Citeseer, 1999.

9. Anaya, I. D. P. *Integrating predictive analysis in self-adaptive pervasive systems*. PhD thesis, Université Rennes 1, 2015.

10. Arsene, O., Dumitrache, I., and Mihu, I. Medicine expert system dynamic bayesian network and ontology based. *Expert Systems with Applications*, 38 (12):15253 – 15261, 2011. ISSN 0957-4174. doi: http://dx.doi.org/10.1016/j.eswa.2011.05.074. URL http://www.sciencedirect.com/science/article/pii/S095741741100858X.

11. Avancha, S., Patel, C., and Joshi, A. Ontology-driven adaptive sensor networks. In *MobiQuitous*, volume 4, pages 194–202, 2004.

12. Babcock, B., Babu, S., Datar, M., Motwani, R., and Widom, J. Models and issues in data stream systems. In *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 1–16. ACM, 2002.

13. Babu, S. and Widom, J. Continuous queries over data streams. *ACM Sigmod Record*, 30(3):109–120, 2001.

14. Baecker, R. M. Picture-driven animation. In *Proceedings of the May 14-16, 1969, spring joint computer conference*, pages 273–288. ACM, 1969.

15. Baecker, R. M. *Readings in Human-Computer Interaction: toward the year 2000*. Morgan Kaufmann, 2014.

16. Barbieri, D. F., Braga, D., Ceri, S., VALLE, E. D., and Grossniklaus, M. C-sparql: a continuous query language for rdf data streams. *International Journal of Semantic Computing*, 4(01):3–25, 2010.

17. Barnaghi, P., Meissner, S., Presser, M., and Moessner, K. Sense and sens' ability: Semantic data modelling for sensor networks. In *Conference Proceedings of ICT Mobile Summit 2009*, 2009.

18. Bashar, A., Parr, G., McClean, S., Scotney, B., and Nauck, D. Bard: A novel application of bayesian reasoning for proactive network management. 2009.

19. Berger, J. O. *Statistical decision theory and Bayesian analysis*. Springer Science & Business Media, 2013.

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

21. Bezerra, C., Freitas, F., and Santana, F. Evaluating ontologies with competency questions. In *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2013 IEEE/WIC/ACM International Joint Conferences on*, volume 3, pages 284–285. IEEE, 2013.

22. Boland, J., Burdett, R., and Edwards, C. Optimisation techniques for planning automatic under frequency load shedding in new zealand's power system. *ANZIAM Journal*, 57(0):1–37, 2016. ISSN 1446-8735. URL http://journal.austms.org.au/ojs/index.php/ANZIAMJ/article/view/10015.

23. Bonnet, P., Gehrke, J., and Seshadri, P. Towards sensor database systems. In *International Conference on Mobile Data Management*, pages 3–14. Springer, 2001.

24. Boroojeni, K. G., Amini, M. H., Bahrami, S., Iyengar, S., Sarwat, A. I., and Karabasoglu, O. A novel multi-time-scale modeling for electric power demand forecasting: From short-term to medium-term horizon. *Electric Power Systems Research*, 142:58 – 73, 2017. ISSN 0378-7796. doi: http://dx.doi.org/10.1016/j.epsr.2016.08.031. URL http://www.sciencedirect.com/science/article/pii/S0378779616303352.

25. Borst, W. N. *Construction of engineering ontologies for knowledge sharing and reuse*. Universiteit Twente, 1997.

26. Botts, M., Percivall, G., Reed, C., and Davidson, J. Ogc® sensor web enablement: Overview and high level architecture. In *International conference on GeoSensor Networks*, pages 175–190. Springer, 2006.

27. Bousdekis, A., Papageorgiou, N., Magoutas, B., Apostolou, D., and Mentzas, G. A probabilistic model for context-aware proactive decision making. In *2016 7th International Conference on Information, Intelligence, Systems Applications (IISA)*, pages 1–6, July 2016. doi: 10.1109/IISA.2016.7785339.

28. Box, G. E. and Jenkins, G. M. Time series analysis forecasting and control. Technical report, DTIC Document, 1970.

29. Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

30. Breiman, L. Random forests. *Machine learning*, 45(1):5–32, 2001.

31. Brickley, D. and Guha, R. Rdf schema 1.1. w3c recommendation (25 february 2014). *World Wide Web Consortium*, 2014.

32. Brown, C. E. and Fingas, M. F. New space-borne sensors for oil spill response. In *International Oil Spill Conference*, volume 2001, pages 911–916. American Petroleum Institute, 2001.

33. Brown, S. et al. Indoor air quality in australian buildings. *Chemistry in Australia*, 71(11):6, 2004.

34. Calbimonte, J.-P. *Ontology-based access to sensor data streams*. PhD thesis, Informatica, 2013.

35. Calbimonte, J.-P., Corcho, O., and Gray, A. J. Enabling ontology-based access to streaming data sources. In *International Semantic Web Conference*, pages 96–111. Springer, 2010.

36. Campbell, J., Gibbons, P. B., Nath, S., Pillai, P., Seshan, S., and Sukthankar, R. Irisnet: an internet-scale architecture for multimedia sensors. In *Proceedings of the 13th annual ACM international conference on Multimedia*, pages 81–88. ACM, 2005.

37. Cardoso, J. The semantic web vision: Where are we? *IEEE Intelligent systems*, 22(5), 2007.

38. Cassandra, A. R. A survey of pomdp applications. In *Working notes of AAAI 1998 fall symposium on planning with partially observable Markov decision processes*, page Vol. 1724, 1998.

39. Chai, J. and Liu, J. N. K. An ontology-driven framework for supporting complex decision process. *CoRR*, abs/1107.2997, 2011. URL http://arxiv.org/abs/1107.2997.

40. Choi, H.-S. and Rhee, W.-S. Distributed semantic sensor web architecture. In *TENCON 2012-2012 IEEE Region 10 Conference*, pages 1–6. IEEE, 2012.

41. Christianini, N. and Shawe-Taylor, J. Support vector machines. *Cambridge, UK: Cambridge University Press*, 93(443):935–948, 2000.

42. Compton, M., Henson, C., Lefort, L., Neuhaus, H., and Sheth, A. A survey of the semantic specification of sensors. In *Proceedings of the 2nd International Conference on Semantic Sensor Networks-Volume 522*, pages 17–32. CEUR-WS. org, 2009.

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

44. Corcho, O. and García-Castro, R. Five challenges for the semantic sensor web. *Semantic Web*, 1(1, 2):121–125, 2010.

45. Cox, S. J. Time ontology extended for non-gregorian calendar applications. *Semantic Web*, 7(2):201–209, 2016.

46. Cyganiak, R., Wood, D., and Lanthaler, M. Rdf 1.1 concepts and abstract syntax. *W3C Recommendation*, 25:1–8, 2014.

47. Delin, K. A. and Jackson, S. P. Sensor web: A new instrument concept. In *Symposium on Integrated Optics*, pages 1–9. International Society for Optics and Photonics, 2001.

48. Della Valle, E., Ceri, S., Barbieri, D. F., Braga, D., and Campi, A. A first step towards stream reasoning. In *Future Internet Symposium*, pages 72–81. Springer, 2008.

49. Diallo, O., Rodrigues, J. J., and Sene, M. Real-time data management on wireless sensor networks: a survey. *Journal of Network and Computer Applications*, 35(3):1013–1021, 2012.

50. Ding, H., Trajcevski, G., Scheuermann, P., Wang, X., and Keogh, E. Querying and mining of time series data: experimental comparison of representations and distance measures. *Proceedings of the VLDB Endowment*, 1(2):1542–1552, 2008.

51. Dongre, P. B. and Malik, L. G. A review on real time data stream classification and adapting to various concept drift scenarios. In *Advance Computing Conference (IACC), 2014 IEEE International*, pages 533–537, 2014. doi: 10.1109/IAdCC.2014.6779381.

52. Doran, J. W. and Zeiss, M. R. Soil health and sustainability: managing the biotic component of soil quality. *Applied soil ecology*, 15(1):3–11, 2000.

53. Dubois, D. and Perny, P. A review of fuzzy sets in decision sciences: Achievements, limitations and perspectives. *Multiple Criteria Decision Analysis*, page 637, 2016.

54. Economou, T., Stephenson, D. B., Rougier, J. C., Neal, R. A., and Mylne, K. R. On the use of bayesian decision theory for issuing natural hazard warnings. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 472(2194), 2016. ISSN 1364-5021. doi: 10.1098/rspa.2016.0295. URL http://rspa.royalsocietypublishing.org/content/472/2194/20160295.

55. Eid, M., Liscano, R., and El Saddik, A. A universal ontology for sensor networks data. In *2007 IEEE International Conference on Computational Intelligence for Measurement Systems and Applications*, pages 59–62. IEEE, 2007.

56. Engel, Y. and Etzion, O. Towards proactive event-driven computing. In *Proceedings of the 5th ACM international conference on Distributed event-based system*, pages 125–136. ACM, 2011.

57. Engel, Y., Etzion, O., and Feldman, Z. A basic model for proactive event-driven computing. In *Proceedings of the 6th ACM international conference on distributed event-based systems*, pages 107–118. ACM, 2012.

58. Fernández-López, M., Gómez-Pérez, A., and Juristo, N. Methontology: from ontological art towards ontological engineering. 1997.

59. Foster, I. Human-machine symbiosis, 50 years on. *arXiv preprint arXiv:0712.2255*, 2007.

60. Friedman, N., Geiger, D., and Goldszmidt, M. Bayesian network classifiers. *Machine learning*, 29(2-3):131–163, 1997.

61. Fulcher, B. D. and Jones, N. S. Highly comparative feature-based time-series classification. *IEEE Transactions on Knowledge and Data Engineering*, 26 (12):3026–3037, 2014.

62. Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, 46(4): 44, 2014.

63. Ganek, A. G. and Corbi, T. A. The dawning of the autonomic computing era. *IBM systems Journal*, 42(1):5–18, 2003.

64. Gibbons, P. B., Karp, B., Ke, Y., Nath, S., and Seshan, S. Irisnet: An architecture for a worldwide sensor web. *IEEE pervasive computing*, 2(4): 22–33, 2003.

65. Grau, B. C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., and Sattler, U. Owl 2: The next step for owl. *Web Semantics: Science, Services and Agents on the World Wide Web*, 6(4):309–322, 2008.

66. Gray, A. J., Sadler, J., Kit, O., Kyzirakos, K., Karpathiotakis, M., Calbimonte, J.-P., Page, K., García-Castro, R., Frazer, A., Galpin, I., et al. A semantic sensor web for environmental decision support applications. *Sensors*, 11(9): 8855–8887, 2011.

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

68. Grüninger, M. and Fox, M. S. The role of competency questions in enterprise engineering. In Rolstadås, A., editor, *Benchmarking — Theory and Practice*, pages 22–31. Springer US, Boston, MA, 1995. ISBN 978-0-387-34847-6. doi: 10.1007/978-0-387-34847-6_3. URL http://dx.doi.org/10.1007/978-0-387-34847-6_3.

69. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

70. Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., and Scholkopf, B. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4):18–28, July 1998. ISSN 1094-7167. doi: 10.1109/5254.708428.

71. Henrion, M., Breese, J. S., and Horvitz, E. J. Decision analysis and expert systems. *AI magazine*, 12(4):64, 1991.

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

73. Horn, P. Autonomic computing: Ibm\'s perspective on the state of information technology. 2001.

74. Horvitz, E. J., Breese, J. S., and Henrion, M. Decision theory in expert systems and artificial intelligence. *International journal of approximate reasoning*, 2(3): 247–302, 1988.

75. Huebscher, M. C. and McCann, J. A. A survey of autonomic computing—degrees, models, and applications. *ACM Computing Surveys (CSUR)*, 40(3):7, 2008.

76. Huizenga, C., Abbaszadeh, S., Zagreus, L., and Arens, E. A. Air quality and thermal comfort in office buildings: results of a large indoor environmental quality survey. *Proceeding of Healthy Buildings 2006*, 3, 2006.

77. IBM, E. An architectural blueprint for autonomic computing. *IBM Publication*, 2003.

78. Jafta, N., Batterman, S. A., Gqaleni, N., Naidoo, R. N., and Robins, T. G. Characterization of allergens and airborne fungi in low and middle-income homes of primary school children in durban, south africa. *American journal of industrial medicine*, 55(12):1110–1121, 2012.

79. James, P., Troped, P. J., and Laden, F. Chapter 50 - the impact of the built environment on health. In Goldman, M. B., Troisi, R., and Rexrode, K. M., editors, *Women and Health (Second Edition)*, pages 753 – 763. Academic Press, second edition edition, 2013. ISBN 978-0-12-384978-6. doi: https://doi.org/10.1016/B978-0-12-384978-6.00050-9. URL https://www.sciencedirect.com/science/article/pii/B9780123849786000509.

80. Janowicz, K. and Compton, M. 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-Volume 668*, pages 64–78. CEUR-WS. org, 2010.

81. Jiang, Y., Li, K., Tian, L., Piedrahita, R., Yun, X., Mansata, O., Lv, Q., Dick, R. P., Hannigan, M., and Shang, L. Maqs: a personalized mobile sensing system for indoor air quality monitoring. In *Proceedings of the 13th international conference on Ubiquitous computing*, pages 271–280. ACM, 2011.

82. Kane, M. J., Price, N., Scotch, M., and Rabinowitz, P. Comparison of arima and random forest time series models for prediction of avian influenza h5n1 outbreaks. *BMC Bioinformatics*, 15(1):1–9, 2014. ISSN 1471-2105. doi: 10.1186/1471-2105-15-276. URL http://dx.doi.org/10.1186/1471-2105-15-276.

83. Kaur, G. and Chhabra, A. Improved j48 classification algorithm for the prediction of diabetes. *International Journal of Computer Applications*, 98(22), 2014.

84. Keogh, E. and Kasetty, S. On the need for time series data mining benchmarks: a survey and empirical demonstration. *Data Mining and knowledge discovery*, 7(4):349–371, 2003.

85. Kephart, J. O. and Chess, D. M. The vision of autonomic computing. *Computer*, 36(1):41–50, 2003.

86. Khashei, M., Bijari, M., and Hejazi, S. R. Combining seasonal arima models with computational intelligence techniques for time series forecasting. *Soft Computing*, 16(6):1091–1105, 2012. ISSN 1433-7479. doi: 10.1007/s00500-012-0805-9. URL http://dx.doi.org/10.1007/s00500-012-0805-9.

87. Klarman, S. and Meyer, T. Prediction and explanation over dl-lite data streams. In *International Conference on Logic for Programming Artificial Intelligence and Reasoning*, pages 536–551. Springer, 2013.

88. Klyne, G. and Carroll, J. J. Resource description framework (rdf): Concepts and abstract syntax. 2006.

89. Kohavi, R. and Sommerfield, D. Targeting business users with decision table classifiers. In *KDD*, pages 249–253, 1998.

90. Krämer, J. and Seeger, B. *A temporal foundation for continuous queries over data streams*. Univ., 2004.

91. Lampe, M., Strassner, M., and Fleisch, E. A ubiquitous computing environment for aircraft maintenance. In *Proceedings of the 2004 ACM symposium on Applied computing*, pages 1586–1592. ACM, 2004.

92. Lécué, F. and Pan, J. Z. Predicting knowledge in an ontology stream. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence*, pages 2662–2669. AAAI Press, 2013.

93. Lécué, F., Tucker, R., Bicer, V., Tommasi, P., Tallevi-Diotallevi, S., and Sbodio, M. Predicting severity of road traffic congestion using semantic web technologies. In *European Semantic Web Conference*, pages 611–627. Springer, 2014.

94. Liaw, A. and Wiener, M. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

95. Licklider, J. C. Man-computer symbiosis. *IRE transactions on human factors in electronics*, (1):4–11, 1960.

96. López-Iñesta, E., Grimaldo, F., and Arevalillo-Herráez, M. Comparing feature-based and distance-based representations for classification similarity learning. In *CCIA*, pages 23–32, 2014.

97. Lu, H. and Liu, H. Decision tables: Scalable classification exploring rdbms capabilities. In *Proceedings of the 26th International Conference on Very Large Data Bases, VLDB'00*, page 373, 2000.

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

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

100. Medvidovic, N. and Taylor, R. N. Software architecture: foundations, theory, and practice. In *Proceedings of the 32nd ACM/IEEE International Conference on Software Engineering-Volume 2*, pages 471–472. ACM, 2010.

101. Mintz, D. Technical assistance document for the reporting of daily air quality-the air quality index (aqi). *Tech. Research Triangle Park, US Environmental Protection Agency*, 2009.

102. Mohamed, M., Arkady, Z., and Shonali, K. Data stream mining. *M. Oded, R. Lior. Data Mining and Knowledge Discovery Handbook. New York: Springer*, page 761, 2010.

103. Moodley, D. and Simonis, I. A new architecture for the sensor web: The swap framework. In *ISWC*, page 5th, 2006.

104. Moodley, D. and Tapamo, J. R. A semantic infrastructure for a knowledge driven sensor web. In *Proceedings of the 4th International Conference on Semantic Sensor Networks-Volume 839*, pages 39–54. CEUR-WS. org, 2011.

105. Moodley, D., Terhorst, A., Simonis, I., Mcferren, G., and Van den Bergh, F. Using the sensor web to detect and monitor the spread of wild fires. 2006.

106. Moodley, D., Simonis, I., and Tapamo, J. R. An architecture for managing knowledge and system dynamism in the worldwide sensor web. *International Journal on Semantic Web and Information Systems (IJSWIS)*, 8(1):64–88, 2012.

107. Moreno, G. A., Cámara, J., Garlan, D., and Schmerl, B. Efficient decision-making under uncertainty for proactive self-adaptation. In *Autonomic Computing (ICAC), 2016 IEEE International Conference on*, pages 147–156. IEEE, 2016.

108. Mozaffari, L., Mozaffari, A., and Azad, N. L. Vehicle speed prediction via a sliding-window time series analysis and an evolutionary least learning machine: a case study on san francisco urban roads. *Engineering Science and Technology, an International Journal*, 18(2):150–162, 2015.

109. Musen, M. A. Protégé ontology editor. *Encyclopedia of Systems Biology*, pages 1763–1765, 2013.

110. Muthukrishnan, S. *Data streams: Algorithms and applications*. Now Publishers Inc, 2005.

111. Nagasawa, K., Upshaw, C. R., Rhodes, J. D., Holcomb, C. L., Walling, D. A., and Webber, M. E. Data management for a large-scale smart grid demonstration project in austin, texas. In *ASME 2012 6th International Conference on Energy Sustainability collocated with the ASME 2012 10th International Conference on Fuel Cell Science, Engineering and Technology*, pages 1027–1031. American Society of Mechanical Engineers, 2012.

112. Naidoo, R. N., Robins, T. G., Batterman, S., Mentz, G., and Jack, C. Ambient pollution and respiratory outcomes among schoolchildren in durban, south africa. *South African Journal of Child Health*, 7(4):127–134, 2013.

113. Nath, S., Deshpande, A., Ke, Y., Gibbons, P. B., Karp, B., and Seshan, S. Irisnet: An architecture for internet-scale sensing services. In *Proceedings of the 29th international conference on Very large data bases-Volume 29*, pages 1137–1140. VLDB Endowment, 2003.

114. Nieto, M. A. M. An overview of ontologies. *Universidad De Las Américas Puebla, Interactive and Cooperative Technologies Lab.*, 2003.

115. Pantelopoulos, A. and Bourbakis, N. G. A survey on wearable sensor-based systems for health monitoring and prognosis. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(1):1–12, 2010.

116. Pentland, A. Looking at people: Sensing for ubiquitous and wearable computing. *IEEE Transactions on Pattern analysis and machine intelligence*, 22(1): 107–119, 2000.

117. Percivall, G. and Reed, C. Ogc sensor web enablement standards. *Sensors & Transducers Journal*, 71(9):698–706, 2006.

118. Perumal, T., Sulaiman, M. N., Mustapha, N., Shahi, A., and Thinaharan, R. Proactive architecture for internet of things (iots) management in smart homes. In *2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE)*, pages 16–17, Oct 2014. doi: 10.1109/GCCE.2014.7031347.

119. Pitarma, R., Marques, G., and Caetano, F. Monitoring indoor air quality to improve occupational health. In *New Advances in Information Systems and Technologies*, pages 13–21. Springer, 2016.

120. Powers, D. M. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. 2011.

121. Pulido, J., Ruiz, M., Herrera, R., Cabello, E., Legrand, S., and Elliman, D. Ontology languages for the semantic web: A never completely updated review. *Knowledge-Based Systems*, 19(7):489–497, 2006.

122. Quinlan, J. R. *C4.5: programs for machine learning*. Elsevier, 2014.

123. Rens, G. Speeding up online pomdp planning. In *Proceedings of the International Conference on Agents and Artificial Intelligence-Volume 2*, pages 241–246. SCITEPRESS-Science and Technology Publications, Lda, 2015.

124. Rens, G. and Moodley, D. A hybrid pomdp-bdi agent architecture with online stochastic planning and plan caching. *Cognitive Systems Research*, 43:1 – 20, 2017. ISSN 1389-0417. doi: http://dx.doi.org/10.1016/j.cogsys.2016.12.002. URL www.sciencedirect.com/science/article/pii/S1389041716300870.

125. Rens, G., Meyer, T., and Lakemeyer, G. A logic for reasoning about decision-theoretic projections. In *International Conference on Agents and Artificial Intelligence*, pages 79–99. Springer, 2015.

126. Rhodes, J. D., Upshaw, C. R., Harris, C. B., Meehan, C. M., Walling, D. A., Navrátil, P. A., Beck, A. L., Nagasawa, K., Fares, R. L., Cole, W. J., et al. Experimental and data collection methods for a large-scale smart grid deployment: Methods and first results. *Energy*, 65:462–471, 2014.

127. Rospocher, M. and Serafini, L. Ontology-centric decision support. In *Proceedings of the 2012 International Conference on Semantic Technologies Meet Recommender Systems & Big Data - Volume 919*, SeRSy'12, pages 61–72, Aachen, Germany, Germany, 2012. CEUR-WS.org. URL http://dl.acm.org/citation.cfm?id=2887638.2887644.

128. Rospocher, M. and Serafini, L. An ontological framework for decision support. In *Joint International Semantic Technology Conference*, pages 239–254. Springer, 2012.

129. Russomanno, D. J., Kothari, C. R., and Thomas, O. A. Building a sensor ontology: A practical approach leveraging iso and ogc models. In *IC-AI*, pages 637–643. Citeseer, 2005.

130. Saad, S. M., Andrew, A. M., Shakaff, A. Y. M., Saad, A. R. M., Kamarudin, A. M. Y. ., and Zakaria, A. Classifying sources influencing indoor air quality (iaq) using artificial neural network (ann). *Sensors*, 15(5):11665, 2015. ISSN 1424-8220. doi: 10.3390/s150511665. URL http://www.mdpi.com/1424-8220/15/5/11665.

131. Salovaara, A. and Oulasvirta, A. Six modes of proactive resource management: a user-centric typology for proactive behaviors. In *Proceedings of the third Nordic conference on Human-computer interaction*, pages 57–60. ACM, 2004.

132. Semple, S., Ibrahim, A. E., Apsley, A., Steiner, M., and Turner, S. Using a new, low-cost air quality sensor to quantify second-hand smoke (shs) levels in homes. *Tobacco control*, pages tobaccocontrol–2013, 2013.

133. Serra, J. and Arcos, J. L. An empirical evaluation of similarity measures for time series classification. *Knowledge-Based Systems*, 67:305–314, 2014.

134. Shadbolt, N., Berners-Lee, T., and Hall, W. The semantic web revisited. *IEEE intelligent systems*, 21(3):96–101, 2006.

135. Sheth, A., Henson, C., and Sahoo, S. S. Semantic sensor web. *IEEE Internet computing*, 12(4):78–83, 2008.

136. Shneidman, J., Pietzuch, P., Ledlie, J., Roussopoulos, M., Seltzer, M., and Welsh, M. Hourglass: An infrastructure for connecting sensor networks and applications. Technical report, Harvard Technical Report TR-21-04, Cambridge, MA, 2004.

137. Simonis, I. and Echterhoff, J. Geoss and the sensor web. In *GEOSS Sensor Web Workshop Report*, 2008.

138. Sokolova, M. and Lapalme, G. A systematic analysis of performance measures for classification tasks. *Information Processing & Management*, 45(4):427–437, 2009.

139. Spreitzer, M. and Theimer, M. *Providing location information in a ubiquitous computing environment (panel session)*, volume 27. ACM, 1994.

140. Srinivasan, S., O'Fallon, L. R., and Dearry, A. Creating healthy communities, healthy homes, healthy people: initiating a research agenda on the built environment and public health. *American journal of public health*, 93(9):1446–1450, 2003.

141. Steinle, S., Reis, S., Sabel, C. E., Semple, S., Twigg, M. M., Braban, C. F., Leeson, S. R., Heal, M. R., Harrison, D., Lin, C., et al. Personal exposure monitoring of pm 2.5 in indoor and outdoor microenvironments. *Science of the Total Environment*, 508:383–394, 2015.

142. Sterritt, R. and Bustard, D. W. Autonomic computing-a means of achieving dependability? 2003.

143. Tahir, M. F., ul Hassan, T., and Saqib, M. A. Optimal scheduling of electrical power in energy-deficient scenarios using artificial neural network and bootstrap aggregating. *International Journal of Electrical Power & Energy Systems*, 83:49 – 57, 2016. ISSN 0142-0615. doi: http://dx.doi.org/10.1016/j.ijepes.2016.03.046. URL http://www.sciencedirect.com/science/article/pii/S0142061516305749.

144. Tan, P., Steinbach, M., and Kumar, V. Classification: Basic concepts, decision trees, and model evaluation in introduction to data mining. ed. *AddisonWesley*, page 769, 2005.

145. Taniar, D. *Web Semantics & Ontology*. Igi Global, 2006.

146. Tennenhouse, D. Proactive computing. *Communications of the ACM*, 43(5): 43–50, 2000.

147. Terry, D., Goldberg, D., Nichols, D., and Oki, B. *Continuous queries over append-only databases*, volume 21. ACM, 1992.

148. Ulbrich, S. and Maurer, M. Probabilistic online pomdp decision making for lane changes in fully automated driving. In *Intelligent Transportation Systems-(ITSC), 2013 16th International IEEE Conference on*, pages 2063–2067. IEEE, 2013.

149. Vafaeipour, M., Rahbari, O., Rosen, M. A., Fazelpour, F., and Ansarirad, P. Application of sliding window technique for prediction of wind velocity time series. *International Journal of Energy and Environmental Engineering*, 5(2-3):1–7, 2014.

150. VanSyckel, S. and Becker, C. A survey of proactive pervasive computing. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication*, pages 421–430. ACM, 2014.

151. Vasimalla, K. A survey on time series data mining. *International Journal of Innovative Research in Computer and Communication Engineering (IJIRCCE)*, 2:170–179, 2014.

152. Voyant, C., Nivet, M. L., Paoli, C., Muselli, M., and Notton, G. Meteorological time series forecasting based on mlp modelling using heterogeneous transfer functions. *Journal of Physics: Conference Series*, 574(1):012064, 2015. URL http://stacks.iop.org/1742-6596/574/i=1/a=012064.

153. Wang, Y. and Cao, K. A proactive complex event processing method for large-scale transportation internet of things. *International Journal of Distributed Sensor Networks*, 2014, 2014.

154. Wanner, L., Rospocher, M., Vrochidis, S., Johansson, L., Bouayad-Agha, N., Casamayor, G., Karppinen, A., Kompatsiaris, I., Mille, S., Moumtzidou, A., and Serafini, L. Ontology-centered environmental information delivery for personalized decision support. *Expert Syst. Appl.*, 42(12):5032–5046, July 2015. ISSN 0957-4174. doi: 10.1016/j.eswa.2015.02.048. URL http://dx.doi.org/10.1016/j.eswa.2015.02.048.

155. Want, R., Pering, T., and Tennenhouse, D. Comparing autonomic and proactive computing. *IBM Systems Journal*, 42(1):129–135, 2003.

156. Weiser, M. The computer for the 21st century. *Scientific american*, 265(3):94–104, 1991.

157. WHO. *Air quality guidelines: global update 2005: particulate matter, ozone, nitrogen dioxide, and sulfur dioxide.* World Health Organization, 2006.

158. WHO. *WHO guidelines for indoor air quality: selected pollutants.* World Health Organization, 2010. ISBN 978 92 890 0213 4.

159. Witten, I. H., Frank, E., and Hall, M. A. Data mining: Practical machine learning tools and techniques. 2011.

160. Wkatróbski, J. An ontology supporting multiple-criteria decision analysis method selection. In *Intelligent Decision Technologies 2016*, pages 89–99. Springer, 2016.

161. Xi, X., Keogh, E., Shelton, C., Wei, L., and Ratanamahatana, C. A. Fast time series classification using numerosity reduction. In *Proceedings of the 23rd international conference on Machine learning*, pages 1033–1040. ACM, 2006.

162. Xing, Z., Pei, J., and Keogh, E. A brief survey on sequence classification. *ACM Sigkdd Explorations Newsletter*, 12(1):40–48, 2010.

163. Yanosky, J. D., Williams, P. L., and MacIntosh, D. L. A comparison of two direct-reading aerosol monitors with the federal reference method for pm 2.5 in indoor air. *Atmospheric Environment*, 36(1):107–113, 2002.

164. Ye, J., Dobson, S., and McKeever, S. Situation identification techniques in pervasive computing: A review. *Pervasive and mobile computing*, 8(1):36–66, 2012.

165. Ye, L. and Keogh, E. Time series shapelets: A new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '09, pages 947–956, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9. doi: 10.1145/1557019.1557122. URL http://doi.acm.org/10.1145/1557019.1557122.

166. Yu, T.-C. and Lin, C.-C. An intelligent wireless sensing and control system to improve indoor air quality: monitoring, prediction, and preaction. *International Journal of Distributed Sensor Networks*, 2015:144, 2015.

167. Zampolli, S., Elmi, I., Ahmed, F., Passini, M., Cardinali, G., Nicoletti, S., and Dori, L. An electronic nose based on solid state sensor arrays for low-cost indoor air quality monitoring applications. *Sensors and Actuators B: Chemical*, 101(1):39–46, 2004.

168. Zhang, G. P. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.

169. Zhao, L., Ichise, R., Sasaki, Y., Liu, Z., and Yoshikawa, T. Fast decision making using ontology-based knowledge base. In *Intelligent Vehicles Symposium (IV), 2016 IEEE*, pages 173–178. IEEE, 2016.

170. Zhao, Y. and Hasan, Y. A. Fine particulate matter concentration level prediction by using tree-based ensemble classification algorithms. *International Journal of Advanced Computer Science and Applications (IJACSA)*, 4(5), 2013.

171. Zheng, H.-T., Kang, B.-Y., and Kim, H.-G. An ontology-based bayesian network approach for representing uncertainty in clinical practice guidelines. In *Uncertainty Reasoning for the Semantic Web I*, pages 161–173. Springer, 2008.

172. Zhu, Y., Smith, T. J., Davis, M. E., Levy, J. I., Herrick, R., and Jiang, H. Comparing gravimetric and real-time sampling of pm2. 5 concentrations inside truck cabins. *Journal of occupational and environmental hygiene*, 8(11):662–672, 2011.