



HAL
open science

A scalable semantic framework for IoT healthcare applications

Rita Zgheib, Stein Kristiansen, Emmanuel Conchon, Thomas Plageman, Vera Goebel, Rémi Bastide

► **To cite this version:**

Rita Zgheib, Stein Kristiansen, Emmanuel Conchon, Thomas Plageman, Vera Goebel, et al.. A scalable semantic framework for IoT healthcare applications. *Journal of Ambient Intelligence and Humanized Computing*, 2020, 10.1007/s12652-020-02136-2 . hal-03695152

HAL Id: hal-03695152

<https://hal.science/hal-03695152>

Submitted on 14 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Scalable Semantic Framework for IoT Healthcare Applications

Rita Zgheib · Stein Kristiansen · Emmanuel Conchon · Thomas Plageman · Vera Goebel · Rémi Bastide

Received: 03 Jul 2019 / Accepted: 30 Apr 2020

Abstract IoT-based systems for early epidemic detection have not been investigated yet in the research. The state-of-the art in sensor technology and activity recognition makes it possible to automatically detect Activities of Daily Living (ADL). Semantic reasoning over ADLs can discover anomalies and symptoms for disorders, hence diseases and epidemics. However, semantic reasoning is computationally rather expensive and therefore unusable for real-time monitoring in large scale applications, like early epidemic detection. To overcome this limitation, this paper proposes a new scalable semantic framework based on several semantic reasoning techniques that are distributed over a semantic middleware. To reduce the number of events to process during the semantic reasoning, a Complex Event Processing (CEP) engine is used to detect abnormal events in ADL and to generate the associated symptom indicators. To demonstrate real-time detection and scalability, the proposed framework integrates a new extension of ADLSim, a discrete event simulator that simulates long-term sequences of ADL.

Keywords Simulated activities · Middleware · ontologies · Epidemic detection

R. Zgheib
Canadian University Dubai, Dubai, United Arab Emirates;
E-mail: rita.zgheib@tud.ac.ae

S. Kristiansen · T. Plageman · V. Goebel
Department of Informatics, University of Oslo, 0316 Oslo, Norway; E-mail: {steikr, plageman, goebel}@ifi.uio.no

E. Conchon
University of Limoges, CNRS, XLIM, UMR 7252, F-87000 Limoges, France; emmanuel.conchon@unilim.fr E-mail: emmanuel.conchon@unilim.fr

R. Bastide
University of Toulouse, IRIT/ISIS, F-81100 Castres, France; E-mail: remi.bastide@irit.fr

1 Introduction

Recent advances in sensor technology, activity recognition for Smart Homes, and Ambient Assisted Living (AAL) promise a wealth of technology support for individuals at home and especially for those with a need for care. However, ambient sensors able to detect whether individuals are healthy or (start to) develop a disease cannot be expected to be readily available in the near future. Activities of Daily Living (ADL) is used for more than 60 years to assess the functional status of people (Katz and et al., 1970) and disease detection like neurodegenerative diseases (Hochgatterer and et al., 2011; Serna and et al., 2007; Lotfi and et al., 2012; Fernández-Llatas and et al., 2011). Furthermore, recent AAL and IoT technologies can automatically detect ADLs (Wu and et al., 2018). Hence, a new IoT-based framework that enables the analysis of ADL is an alternative to ambient health sensors. Analyzing ADLs can reveal symptoms which in turn be used to estimate the probability that an individual has a certain disease. For example, a continuously reduced activity level might indicate fatigue or an unusually high frequency of toilet visits might indicate diarrhea. (Zgheib and et al., 2017) proposed SeMoM, a semantic message oriented middleware architecture for IoT-based monitoring of ADLs. The authors show that semantic interoperability as well as the integration of expert knowledge in an IoT system can be provided through ontologies. For example, the Disease Ontology (DO) (Schriml and et al., 2011) and SYMP (Schriml, 2018) are two ontologies that can be used to describe symptoms and diseases in an IoT system environment.

Thus, even if sensor technology is not yet ready to directly detect whether a human has a disease, it is feasible to automatically detect certain diseases through

continuous ADL analysis to derive symptoms and map these symptoms to diseases using ontologies. Each step of such a chain of data processing tasks brings in a higher level of semantics, thus allowing to eventually derive information on the occurrence of a specific disease. Interestingly, the computational complexity is also increasing step by step in this processing chain. This might not pose a problem when processing the sensor data from a single home, but the more data needs to be processed the bigger is the challenge to perform this in real-time with a given computing infrastructure. Solving this scalability issue is the main challenge that we tackle in this work.

Scalable solutions are important, because the serious health problems of greatest significance today are chronic diseases which are frequent in senior citizens living in retirement homes, or in children living in orphan houses (Thakar and Pandya, 2017). These environments are prone to acquisition and spread of nosocomial infection because residents share the same sources of air, food, water, and health care in a crowded institutional setting (Strausbaugh and et al., 2003). For instance, the elderly population has gained research attention since it is particularly vulnerable to several diseases that could be infectious. Also, epidemic outbreaks of infectious diseases in retirements house increases the mortality rate of the resident population as presented in (Ziakas and et al., 2016).

In this paper, an extension of the SeMoM architecture is proposed to monitor diseases and epidemics. Detecting the outbreak of epidemics means to continuously monitor the health status of large populations and correlate the detected cases of diseases over time. Processing the data from large populations introduces a scalability problem. Especially the last step in the data processing chain, i.e., the use of an ontology to map symptoms to diseases is computationally very costly. Indeed, the time needed for a single invocation of a semantic query (e.g. SPARQL or C-SPARQL query) is correlated with the amount of data it has to process. This paper proposes a solution to the scalability problem, in order to enable real-time epidemic detection in very large populations.

The core idea to solve this problem is to combine different data analysis techniques and semantic reasoning (like SPARQL and C-SPARQL) in such a way that the large amounts of activity data are processed with a system that is dedicated for Complex Event Processing (CEP). For instance, Esper (espertech, 2006) is CEP-based engine that is several orders of magnitude faster than C-SPARQL as shown in (Margara et al., 2018). The reason for this choice of these different technolo-

gies is the trade-off between performance and the ability to perform semantic reasoning.

The paper is organized as follows: in Section 2, a state of the art of the several disease and epidemic detection solutions is presented as well as IoT frameworks in AAL environments. Section 3 presents a detailed description of the new Scalable Semantic Framework for IoT healthcare applications and its application to epidemic detection. In Section 4, the experiments and performance evaluation that demonstrate the scalability of the proposed framework are presented. Finally, Section 5 presents the future work and possible research fields.

2 State of the Art

The core idea of the research work proposed in this paper is to provide an IoT-based solution that tackles two problems. The first problem is related to the disease and epidemic detection in large compounds. The second problem is the scalability, because semantic reasoning techniques present high overhead. This section presents a literature survey of a broad range of research solutions that studied disease detection, IoT frameworks and reasoning techniques.

2.1 From ADL to Epidemic Detection in AAL

Several types of sensors have been proposed as data sources for activity monitoring. They can be classified into multimedia-based (Onofri and et al., 2016) and sensors-based solutions (wearable, smartphone-based or ambient) as presented in (Avci and et al., 2010; Alessandra and et al., 2016). These solutions have been proposed to analyse and parse a stream of data so that human activities can be inferred (Pires and et al., 2016). Human activity has been studied at various levels. Some works focus on simple tracking of persons, others focus on estimating the physical state of persons in the scene and various analyses have been conducted on detecting human actions (simple and complex) (Abdallah and et al., 2018). In particular, an activity detection system in AAL environments automates the recognition of ADL such as eating, grooming, cooking, drinking, and taking medicine.

Going beyond the typical activity detection, some works discussed how to analyze the information related to the detected activities. As presented in (Ryoo and Aggarwal, 2006), the high-level understanding of human activity may enable the detection of the normal behavior and abnormal behavior of persons. For example, fall detection monitoring systems (Lampoltshammer,

2014; Pierleoni and et al., 2015; Yu and et al., 2012) consist on recognizing anomalies in the activities of the person at home. This topic is very well studied due to its importance in providing a safe aging in place. The correlation between lifestyle and health status has been introduced in (Ghayvat and et al., 2018) and disease detection approach has been defined as a possible direction of anomaly detection. In (Perriot and et al., 2014), the authors propose an algorithm to characterize physical and abnormal activities in COPD (Chronic Obstructive Pulmonary Disease) patients. Noury et al. (Noury and et al., 2016) show that abnormal behavior of persons at home reflects the correct functioning of their homeostasis, thus their health and well being. Moreover, the smart environment can help to identify and model progression of neurodegenerative disorders (Hochgatterer and et al, 2011) such as dementia and Alzheimer (Varathanas and et al., 2017; Krishnan and Cook, 2014) by evaluating performances in the execution of ADL such as repeating several times the same activity as shown in (Serna and et al., 2007). In this trend, the solutions proposed in (Lotfi and et al., 2012; Fernández-Llatas and et al., 2011) aim to identify relevant information regarding any anomalous behavior for elderly occupants suffering from dementia.

Anomalies in human behavior such as *not eating* and *not going to toilet regularly* (Hsu and Chen, 2010) can be seen as symptom indicators to diseases such as stomach problems or diarrhea. It plays a prominent role in healthcare analytics and could increase the medical practitioner's knowledge. An expert-like decision could be taken even by a semi-trained individual particularly in response to the medical emergency as described in (Ukil and et al., 2016).

From the other side, people living in crowded conditions (military, cruise ships, dorms) are at high risk of contamination of diseases, as are people living in developing countries who often have a diet that contains contaminated food or water. IoT has been found recently as an effective and proactive solution which provides monitoring of viruses and epidemics. Ebola is one of the deadliest and most infectious viruses which spreads rapidly and affects a large fraction of the population. In (Sareen et al., 2016), a cloud-based architecture is proposed for monitoring and controlling the Ebola disease. It provides automatic categorization of patients into different categories (uninfected, susceptible, exposed, infectious, highly infectious, and recovered). The infected users or regions are identified as involved in the spread of the disease based on the proposed Temporal Network Analysis (TNA). This architecture is based on wireless body area network (WBAN) to collect vital parameters of the body and RFID tech-

nology for capturing the close proximity interactions between users. An extension of this proposal has been provided in (Sareen and et al., 2017) for monitoring the Zika virus which has been declared an emergency by the World Health Organization (WHO) as it has affected newborn babies with microcephaly and neurological disorders.

Despite its major and critical effect, disease and epidemic detection is not investigated enough in the research domain. As previously presented, the work done for epidemic detection is centered on detecting one specific disease/epidemic such as Ebola. This paper covers this topic and an IoT-based framework is proposed to enable the analysis of the daily activities of elderly and the processing of data when deviations of daily activities are observed. These deviations could infer symptoms-indications that reveal a possible disease for the elderly.

2.2 IoT-based Framework Solutions in AAL

Providing an IoT framework consists in defining an architecture that connects together numerous physical sensors, integrates expert knowledge and enables reasoning on data based on defined rules. Middleware solutions have been found to address technical interoperability issues and communication requirements in an IoT environment (Razzaque and et al., 2016). Service Oriented Architecture (SOA) (Wolf and et al., 2010) and Message Oriented Middleware architecture (MOM) have been recognized in the literature to provide an interoperable platform and management services between sensors and applications in AAL (Ambient Assisted Living) context. Especially, MoM (Curry, 2004) architecture with its publish/subscribe paradigm has been put into light for its effectiveness in offering loose coupling communication between IoT components which enforces scalability in such environments.

Ontologies have been proved to provide a formal and logic description for sensors and data in AAL environments (Ziaeeafard and Bergevin, 2015; Meditskos et al., 2016). Specifically, domain applications use ontologies and semantic description as a way to define types, properties, and interrelationships of the entities that exist for a specific domain. In healthcare, many ontologies describe diseases. For instance, SNOMED-CT (Donnelly, 2006) is a well known ontology that represents an advanced terminology and coding system for eHealth. The Disease Ontology (DO) (Kibbe and et al., 2014) represents human diseases for linking biomedical knowledge through disease data. It has been developed in 2011 as a standardized ontology for human disease with the purpose of providing the biomedical community with con-

sistent, reusable and sustainable descriptions of human disease terms. An extended version (Kibbe and et al., 2014) has been presented in 2015. The SYMP (Schriml, 2018) ontology was designed to describe symptoms and signs for illness around the following concept: "A symptom is a perceived change in function, sensation or appearance reported by a patient indicative of a disease". In (Mohammed and et al., 2012), authors propose to align DO and SYMP in order to have a Diseases Symptoms Ontology for Medical Diagnosis. They have added the *has_Symptom* object property to link each disease to its symptoms. Detecting diseases and epidemics in AAL environment requires the integration of domain knowledge and linking symptoms with diseases. For that, in this paper, DO and SYMP ontologies are acquired to provide expert knowledge description as well as conceptualization of the disease domain.

In recent framework approaches, researchers propose (Lohr et al., 2015; Bispo et al., 2015) to combine middleware solutions with semantic techniques, because it provides fully interoperable, flexible, and scalable architectures for IoT applications in AAL context. SMArc (Rodríguez-Molina and et al., 2013), OM2M (Alaya and et al., 2014), and OpenIoT (Kim and Lee, 2014), are examples of these frameworks. Each solution focuses on describing sensors data and metadata in a single semantic format (RDF) using domain specific ontologies. Semantic data is then communicated via a middleware architecture. The common goal of such frameworks is to tackle the heterogeneity and the issues related to the physical infrastructure of sensors in IoT, and to provide important manageability and collaboration between IoT applications. SeMoM is a Semantic Message Oriented Middleware architecture for IoT healthcare applications proposed in (Zgheib and et al., 2017) and applied for activity detection use case. The aim of this architecture is to facilitate sensors data collection and the inference of new knowledge or information needed to detect an event such as diseases of the monitored person. The authors showed the effectiveness of this architecture, but the complexity and overhead of semantic reasoning is still a drawback and affects the scalability. In this paper, this limitation is studied and an extension of SeMoM is proposed to tackle this issue. In the following section, a brief description of SeMoM is presented and the extension is described in the following section

2.2.1 SeMoM architecture

SeMoM (Semantic Message Oriented Middleware) (Zgheib and et al., 2017) is an IoT architecture that has been proposed to deal with the complexity of designing and

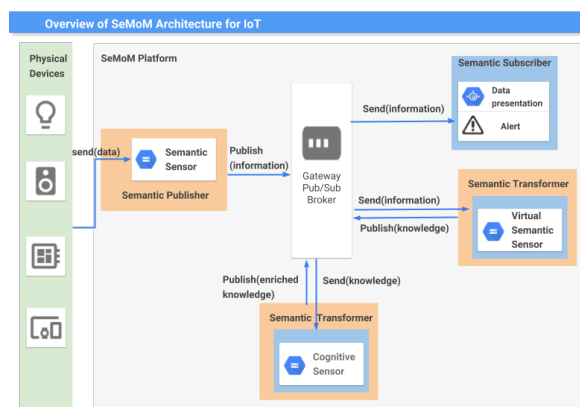


Fig. 1 Overview of the SeMoM architecture for IoT Applications.

managing IoT systems, the heterogeneity of the generated data, the scalability, and the lack of flexibility. SeMoM offers a flexible and interoperable communication system and allows collaboration between IoT applications. It facilitates the integration of novel applications that reason based on information from several data sources. It uses MoM to achieve a loose coupling communication between software components through the publish/subscribe model. Data sources (publishers) and destinations (subscribers) are decoupled from each other and exchange messages based on predefined topics expressed as subscriptions through a message broker.

On top of MoM, there is the semantic layer that provides a semantic enrichment of data and improves the semantic interoperability through an IoT system. The Cognitive Semantic Sensor Network ontology (CoSSN) ontology is responsible for semantic data enrichment. It is a generic ontology that enables describing domain concepts and sensors in the same ontology which is not proposed by other solutions. It can be extended with application domain concepts such as those related to healthcare. It includes conceptual entities from the SSN (Semantic Sensor Network) ontology (Haller and et al., 2018) related to the description of sensor data and metadata and sensors observations.

Based on the combination of semantic concepts with MoM concepts, new components like *semantic sensors*, *virtual semantic sensors* and *cognitive sensors* have been defined (see Figure 1) involving both semantic and communication features.

2.3 Reasoning techniques in AAL

In order to process and reason on sensors data, IoT frameworks should rely on reasoning techniques. Machine Learning, and semantic techniques are leading solutions at the time of writing. As an example for

Machine Learning techniques, authors in (Sareen and et al., 2017), used the Naive Bayesian Network (NBN) to detect the possibly infected users and used Google Maps Web service to provide the geographic positioning system (GPS)- based risk assessment to prevent the outbreak. A clustering analysis technique has been proposed in (Kwok and et al., 2002) to model the normal behavior of elderly and determine the boundaries of all clusters. When an event is observed outside the boundaries, it is viewed as an anomaly. Predictive techniques such as Markov model have also been used in anomaly detection projects such as in MonAMI project (Mainstreaming on Ambient Intelligence) (Novák and et al., 2012). From the other side, Hoque et al. apply semantic rules that explain specific variations of activities in specific scenarios in Holmes (Hoque et al., 2015).

In the ODHMAD (Online Daily Habit Modeling and Anomaly Detection) model (Meng et al., 2017), the authors develop an on-line activity recognition (OAR) algorithm that consists on determining the occurrence of activities by modeling the activation status of sensors. Aran et al. (Aran and et al., 2016) propose a probabilistic spatio-temporal model to summarize and describe natural daily behavior of elderly. Anomalies are then defined as significant changes from the learned behavioral model and detected using a cross-entropy measure. They also propose a semantic layer with the purpose of creating a common format of generated events for different possible sensors.

The most distinguishable feature of the machine learning techniques is their ability to correlate and combine sensors data with location information to extract activities such as running or walking. However, these techniques usually require a long period of learning before being able to predict the activities. Semantic reasoning is gaining more interest recently and a number of systems and languages were produced, aiming at RDF stream processing (Khrouf and et al., 2016). Their advantage pertains to the ability of processing semantic data in a semantic IoT framework. However, their computational complexity affects the scalability and real-time requirements in IoT.

Integrating middleware approaches with semantic technologies creates a suitable platform and environment for IoT applications as shown in (Zgheib and et al., 2019). But the scalability, and real-time requirements remain open challenges for applying semantic methods. In this paper, we propose to combine CEP with semantic techniques and to apply it over a semantic middleware approach for epidemic detection. CEP (Eckert and et al., 2011) encompasses methods, techniques, and tools for processing events and timely detection of complex events. DO and SYMP ontolo-

gies provide mapping between symptoms and diseases, hence they provide the needed domain knowledge that is required in our framework for disease detection. Relying on DO ontology requires the use of the SPARQL (Quilitz and Leser, 2008) query language to retrieve and manipulate data in the DO ontology and the C-SPARQL (Barbieri and et al., 2010) to query Stream of semantic data.

To summarize, many research activities exist to address specific diseases such as Ebola and Zika. However, a thorough search of the relevant literature yielded no proposal on how to achieve a highly scalable solution for semantic reasoning over a large number of events to detect a possible disease/epidemic.

3 Scalable Semantic Framework for early Epidemic Detection

In this section, the methodology adopted to achieve high scalability for a semantic framework for early epidemic detection is described as well as the different proposed components of this framework.

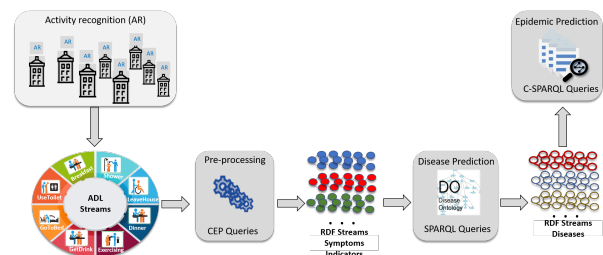


Fig. 2 Overview of the modular process of the Epidemic Detection in nursing home complex.

3.1 Methodology

Figure 2 presents the methodology that has been used as a basis for the framework. This methodology can be presented as follows: **Anomalies in ADL sequences can be analyzed as symptom indicators (SI). The resulting stream of symptoms can be analysed to detect a disease. When the same disease is detected for many apartments/individuals a possible epidemic is occurring.** For example: An abnormal toileting activity can be an indicator to diarrhea symptom; a list of diarrhea, vomiting, and fatigue presents symptoms to gastroenteritis; and when more than 20% of the apartments present the same disease a possible epidemic is detected.

The core idea is to rely on a hybrid solution based on CEP techniques and semantic reasoning on top of SeMoM to process these streams of data. ADL sequences

are generated from each apartment complex, a CEP based tool is used for pre-processing to generate SI that capture abnormalities in activities. SI streams are then semantically enriched in order to apply semantic queries and detect a potential disease. Finally, a potential epidemic is detected based on how many apartments are infected.

3.2 SeMoM extension

In order to detect a possible epidemic, three required information have been defined in the presented methodology: ADL streams, SIs and Diseases. Based on the SeMoM architecture, the main data source for ADL streams are sensors. A scalability analysis of the system requires realistic, large scale datasets, e.i., ADL traces across several months from many apartments. Such traces do not currently exist. The deployment of numerous sensors is prohibitively expensive, the collection of data is constrained by privacy concerns, and the resulting traces typically do not include the anomalies we want to detect in the evaluation of the system. To solve this problem, data is generated with the simulator ADLSim.

Semantic sensor, Virtual semantic sensor, and cognitive sensors have been defined in SeMoM architecture (Section 2.2.1). In this paper, a semantic hierarchy of the different sensors in an IoT environment is defined and presented in Figure 3. In this hierarchy, the smartness of sensors is related to the level of richness of semantic data and the level of knowledge provided by each sensor. Physical and semantic sensors are domain-independent sensors that can be used in any domain, they generate observations that contain respectively sensor output and basic description of sensor outputs. The virtual semantic and cognitive sensors are domain-specific sensors, they embed a software component to interpret and reason on received data, and generate new information as well as new knowledge.

The SeMoM architecture is extended and new components and sensors have been defined based on the previously presented hierarchy. The scalable semantic framework is presented in Figure 4. It is based on four main components. Each one of them is a specialized software component and is responsible to detect/receive data, analyze/reason on them and generate new information. To perform this task each component *subscribes* to the broker by specifying the topic of interest. When a new information is ready, each component publishes it to the middleware by specifying the topic related to the published information. For instance, the topics defined in the epidemic framework are ADL, SI, disease.

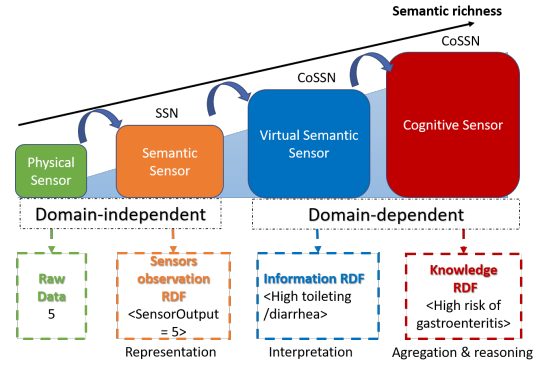


Fig. 3 Overview of the hierarchy of the different connected sensors in IoT.

and epidemic. The four collaborate and exchange information to detect an epidemic as follows:

- ADL component: is a semantic Sensor that use the CoSSN ontology to describe physical sensors observations and publishes them to the broker for the topic ADL. This framework is proposed to be used with implemented physical devices at home. However, due to the load of sufficient ADL datasets for testing and evaluation, physical sensors are replaced by the ADLSim simulator to generates sequences of ADL at each home.
- SI component: a virtual semantic sensor and a CEP-based component that subscribes to ADL topic, detects anomalies in ADL sequences to indicate particular symptoms (SI), and publishes SI streams to the broker.
- Disease component: a cognitive sensor and a SPARQL-based component that subscribes to SI topic, analyzes the anomalies and uses the expert knowledge in the DO ontology to detect a potential disease. It publishes observations containing a possible disease in a specific apartment.
- Epidemic component: a cognitive sensor and a C-SPARQL-based component with the goal to reason on the stream of RDF observations of diseases and detect a potential epidemic when it occurs. Epidemic component subscribes to disease topic and publishes observations to the broker for the topic epidemic.

In this section, the architecture of the framework is presented and the process view of the different components is described. SI component is first described assuming that ADLSim data generator is publishing ADLs sequences. ADLSim and its extensions are detailed later in Section 3.6

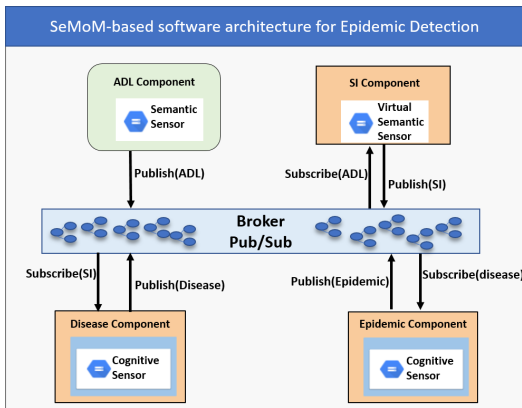


Fig. 4 Overview of the semantic framework to detect the spread of disease in a nursing home complex.

3.3 SI component

The main objective of this work is to detect the outbreak of a disease which is observed in terms of symptoms the disease causes in an infected human. These symptoms are derived from ADL sequences streams which are sent from each apartment to the middleware.

3.3.1 SI Queries: CEP for Symptom Indication

Physical sensors do not have the capability of detecting symptoms. For this reason, the concept of a *Symptom Indicator (SI)* has been created to characterize aspects of activities in a manner that helps to determine the likelihood of the presence of particular symptoms.

Detection of in an ADLs stream is implemented using the Esper CEP library based on the concepts in a previous work on Event Proximity (Kristiansen and et al., 2016). A CEP query for SI is called an *SI query*. SI queries take as input activity sequences, and continuously assess the magnitudes of selected activities in a sliding window of 24 hours, called the *current situation*. Such magnitudes typically include the total duration, frequency, and energy expenditure of activities that are affected by a disease. For instance, gastroenteritis usually results in diarrhea, which is reflected in the duration and frequency of toileting. The current situation exists at two levels: for individual apartments (apartment specific SI) and for the complete apartment complex (complex-wide SI). While complex-wide SI is useful to assess the likelihood of an epidemic, apartment-specific SI is useful to identify the infected individuals.

The output of SI queries is quantified on a scale from one to five, where three represents a normal magnitude of the measured activity, two and one represent low and very low magnitudes, respectively, and four and five represent high and very high magnitudes, respectively. For any given current situation, the SI values

are determined based on the deviations of activity magnitudes from the learned, normal 24-hour magnitudes. Based on the observation of normal activity, we apply maximum likelihood estimation to construct Gaussian distributions that capture the normal magnitudes. An SI value of three is emitted whenever the magnitudes for the current situation lie within ± 1 standard deviation (SD) from the normal mean. SI values of two and four indicate magnitudes that deviate by more than one SD but less than two SD, below and above the normal mean, respectively. Similarly, SI one and five denote magnitudes that deviate by more than two SD above or below the normal mean, respectively.

When applying the same SI threshold for apartment-specific SI, our preliminary experiments show that the resulting SI fluctuate considerably more than the complex-wide SI. Our analysis shows that this is caused by the considerably lower amounts of data available for the current situation. To alleviate this problem, we select wider SI thresholds for apartment-specific SI.

A key advantage of SI queries stems from the possibility to define thresholds for when to emit SI events (called *SI emission thresholds*), in effect providing a facility to generate alarms upon abnormal SI. Since such events are emitted only in abnormal situations, significantly fewer events are generated, and significantly fewer resources are required to analyze the SI, compared to the number of ADLs SI queries analyse. One can adjust the sensitivity of the SI queries emissions by adjusting the threshold, e.g., to emit events for high or low values, or only for very high and low values. In addition, the queries can be configured to emit daily SI summaries (called *24-hour SI statistics*) to enable a continuous search for potential long-term trends.

3.3.2 Semantic Enrichment of SI

As explained in Section 3.3.1, *SI queries* generate SI streams, e.g., "*SI very high*" and "*SI very low*" for particular symptoms when abnormalities occur during activities. In order to integrate the SC in the SeMoM architecture, we embed it as a virtual sensor in the CoSSN ontology. This involves the semantic enrichment of each SI concerning each activity.

For example, a SI for the *ActivityLevel* sent by the CEP component will be semantically described as presented in the Listing 1. In this example, we describe the *SI_Observation126* related to the *observedProperty ActivityLevel* of the elderly in *Apartment 82*. The number of the apartment is added as a *featureOfInterest* linked to the observation by the object property *has-FeatureOfInterest*. This observation has been observed at "1970-03-12T08:14:00:00+10:00", and the time has

been added in the *resultTime* concept. Finally, the value of $SI = 1$ (i.e., "very low") in this case is linked to the observation by the *hasSimpleResult* object property.

After creating the semantic observations using the CoSSN ontology, the SC connects to the broker and publishes the observations in RDF format enabling the use of this information by other components connected to the broker. It publishes seven RDF messages related to the seven SI (see ADLSim section 3.6.1) for each apartment for a specific time when there is an alert. Each published message is labeled with the corresponding SI as the topic of the message. For instance, to send the message in Listing 1, we use *publish("ActivityLevel", SI.Observation126)*.

Listing 1 An example of the semantic observation of the SI "ActivityLevel" sent by ADLSim.

```
<CoSSN/SI.Observation126> a sosa:Observation;
sosa:observedProperty <CoSSN/ActivityLevel>;
sosa:hasFeatureOfInterest <CoSSN/Apartment82>;
sosa:madeBySensor <CoSSN/VirtualSemanticSensor/SI.VirtualSensor82
>;
sosa:resultTime "1970-03-12T08:14:00:00+10:00"^^xsd:dateTime;
sosa:hasSimpleResult "1"^^xsd:int;
```

3.4 Disease Component (DC)

In the proposed solution, the DC subscribes to the SI information to assess if there is a possible disease in a specific apartment. In order to map the semantic observations of SI to a disease, expert knowledge is needed to link a set of symptoms to a specific disease. This framework relies on the proposed aligned ontology in (Mohammed and et al., 2012). DO and SYMP ontologies have been aligned in order to have a Diseases Symptoms Ontology for Medical Diagnosis. The *has_Symptom* object property has been created to link each disease to its symptoms. For example, the gastroenteritis disease has symptoms: diarrhea, vomiting, nausea, and fatigue as shown in Figure 5. In our disease detection model, we rely on the disease/symptom model to detect if there is a possible disease in a specific apartment.

In the CoSSN ontology, the DC is considered as a cognitive sensor that detects the SI from SI queries to publish the corresponding disease to the broker. Between the detection of SI and the detection of diseases,

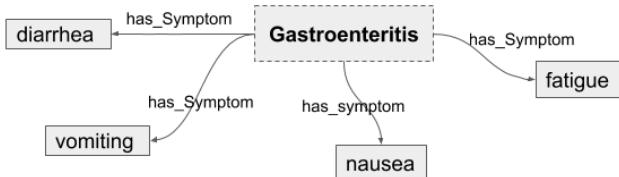


Fig. 5 The disease and symptoms model for gastroenteritis.

there are two main steps. The first one is to map the SI to symptoms and the second is to aggregate these symptoms and to find the corresponding disease. Based on this classification, we can infer symptoms related to the SI. For example, if an SI for the amount of toiletting is "very high", a potential symptom is a *diarrhea*. When an SI for the activity level is "very low", a possible symptom is *fatigue*.

When the DC receives all SI for a specific apartment(inhabitant), it analyzes the values and creates a set of possible symptoms. Then, the Disease Component runs a SPARQL query searching the corresponding disease for the set of symptoms. An excerpt of such query is presented in the Listing 2. It should be noted that the Disease component uses a cache to limit the number of accesses to the ontology and accelerate the simulation. This cache is filled with results of an ontology access for future similar requests.

Listing 2 An excerpt of SPARQL query searching for a specific disease based on set of symptoms.

```
"prefix oio: <http://www.geneontology.org/formats/oboInOwl#>"
+ "prefix owl: <http://www.w3.org/2002/07/owl#>"
+ "prefix rdfs: <http://www.w3.org/2000/01/rdf-schema#>"
+ "prefix doi: <http://purl.obolibrary.org/obo/doi#>"
+ "prefix DOID: <http://purl.obolibrary.org/obo/DOID->"
+ "prefix SYMP: <http://purl.obolibrary.org/obo/SYMP->"
+ "SELECT ?baseLabel ."
+ "WHERE {"
+ "?baseClass rdfs:subClassOf* ?superClass ."
+ "?baseClass rdfs:label ?baseLabel ."

+ "?baseClass rdfs:subClassOf ?symp1 ."
+ "?symp1 owl:Restriction ."
+ "?symp1 owl:onProperty doi:has_symptom ."
+ "?symp1 owl:someValuesFrom " + symptoms.get(0) + " ."

+ "?baseClass rdfs:subClassOf ?symp2 ."
+ "?symp2 owl:Restriction ."
+ "?symp2 owl:onProperty doi:has_symptom ."
+ "?symp2 owl:someValuesFrom " + symptoms.get(1) + " ."

+ "FILTER (?superClass = DOID:4)"
// subclasses of disease
+ "}";
```

When a new disease is found in a specific apartment at a specific time, the DC publishes a new message for the topic "disease". The message contains the same information as a semantic observation described in SSN. An example of such a published message is presented in the Listing 3 where we detected a gastroenteritis in *Apartment 82* at "1970-03-12T08:14:00:00+10:00".

Listing 3 An example of the semantic observations sent by the Disease cognitive sensor.

```
<CoSSN/DiseaseObservation120> a sosa:Observation;
sosa:observedProperty <CoSSN/Disease>;
sosa:hasFeatureOfInterest <CoSSN/Apartment82>;
sosa:madeBySensor <CoSSN/CognitiveSensor/DiseaseCognitiveSensor>;
sosa:resultTime "1970-03-12T08:14:00:00+10:00"^^xsd:dateTime;
sosa:hasSimpleResult "gastroenteritis"^^xsd:string;
```

The DC allows to infer if a specific disease occurred in a specific apartment at a specific time thanks to the DO ontology that provides the needed expert knowledge. However, applying real-time advanced reasoning

on the disease data could infer more relevant information. As an example, the duration of a disease in a specific apartment, the number of infected apartments in a specific duration or a potential epidemic can be inferred by applying the C-SPARQL reasoning technique on the disease data. In the following, we present the Epidemic Component as a C-SPARQL component for real-time reasoning on RDF streams of disease.

3.5 Epidemic Component (EC)

Regarding the Oxford dictionaries, an epidemic is "A widespread occurrence of an infectious disease in a community at a particular time". In this definition there are no crisp, quantitative rules to apply for epidemic detection. Hence, monitoring an epidemic in a medical complex such as a nursing home requires first to define the needs of the medical staff and the expert knowledge. Several possibilities are available at this stage. The first one is the need of the carer to know if there is a spread of a disease in the complex. In such case, we need to track how the number of infected apartments evolves over several days. Another possible requirement arises when a specific disease is already detected in the complex. Hence, we need to detect as soon as possible a potential infection in new apartments. The epidemic component aims to answer the queries of the medical staff and to provide the related knowledge continuously based on the received RDF streams of disease.

For that, the EC collects data by subscribing to the *disease* topic that tags the streaming of diseases sent by the Disease Component. Then, it uses the C-SPARQL reasoning technique to infer the number of infected apartments. The C-SPARQL (Barbieri and et al., 2010) language is an extension of SPARQL that supports continuous queries. The queries are registered over RDF data streams and then are continuously executed.

The knowledge inferred from the C-SPARQL queries defines a new semantic observation that this component can publish to the broker again. The EC plays both roles of publisher and subscriber, so it is described as a cognitive sensor in the CoSSN ontology. It detects the streaming of diseases observations and observes the "Epidemic" property. An example of a possible result for a C-SPARQL query over the last 10 days that looks for apartments where the disease have persisted for five days as shown in Listing 4. The query denotes the first and last days of infection.

Listing 4 An example of a possible result for C-SPARQL query

```
http://www.w3.org/ns/CoSSN/apartment12 "2329200000"^^http://www.w3.org/2001/XMLSchema#long "2674800000"^^http://www.w3.org/2001/XMLSchema#long
```

```
http://www.w3.org/ns/CoSSN/apartment96 "2674800000"^^http://www.w3.org/2001/XMLSchema#long "3020400000"^^http://www.w3.org/2001/XMLSchema#long
http://www.w3.org/ns/CoSSN/apartment95 "2329200000"^^http://www.w3.org/2001/XMLSchema#long "2674800000"^^http://www.w3.org/2001/XMLSchema#long
```

3.6 Epidemic Simulation

Due to the lack of sufficient ADL datasets for testing and evaluation, ADLSim is used in the proposed framework to generate the data.

3.6.1 ADLSim

ADLSim is the main data source and implemented as a semantic sensor in the proposed framework that will publish ADL streams. The onset and duration of performed activities (called activity instances) are simulated based on statistical distributions and rules to handle activity conflicts. It is a discrete event simulator, enabling the experimenter to define arbitrary changes in simulated behavior to occur at any point in simulated time, of any duration and magnitude, and either gradually or instantly. It can also produce highly realistic results when the activity models are parametrised based on real world traces (Kristiansen and et al., 2018). In this work, we use one of the most widely cited publicly available traces, i.e., the one published by Kasteren et al. in (Van Kasteren and et al., 2008). The included activities are selected based on the Katz independence index (Katz and et al., 1970). This is a widely acknowledged measure in health care of the independence of people that live alone, and is particularly relevant to assess the independence of elderly people. Thus, these activities consist of the most essential ADL, and are therefore exactly the set of activities we wish to include in our simulation since most people perform these activities every day. These activities are named (IDs in parenthesis) *leave house* (1), *use toilet* (2), *take shower* (3), *go to bed* (4), *prepare breakfast* (5), *prepare dinner* (6) and *get drink* (7). Their names are accurate indicators of the activity that is performed. Consult (Katz and et al., 1970; Van Kasteren and et al., 2008) for more detailed description of the involved activities and how the real world traces are captured, and (Kristiansen and et al., 2018) for details on the parametrization of the corresponding activity models.

To use ADLSim for the evaluation of the proposed solution, it is extended with multi-apartment support as well as epidemic and disease models.

3.6.2 Multi-apartment Simulation

ADLSim was originally designed to simulate activities within a single apartment. In the current work, it is therefore necessary to extend ADLSim to support many-apartment simulations. We follow a two-stage simulation approach. In Stage 1, we perform N separate simulations of activity within N apartments with the same simulated duration but with different random seeds and store the activity traces for all apartments within one trace file. All activity instances in the trace file are prefixed with the ID of the apartment in which they were performed. In Stage 2, a single simulation run is performed, taking as input the trace files from Stage 1. Here, all events from all N simulations are executed in parallel, simulating the activity in N apartments simultaneously. Note that this solution cannot simulate the effect of interaction between separate apartments.

Infections like gastroenteritis are typically accompanied by a reduction in activity level. It is therefore useful to also extend ADLSim with the ability to simulate energy use per activity per time unit, that can be temporarily reduced during simulation to simulate reductions in activity level. The ADL traces used to parametrize ADLSim lacks the necessary metrics and values for energy expenditure. Fortunately, this is not needed to demonstrate that the queries can detect long-term, significant changes in activity levels. Instead of using traditional metrics like calorie consumption, ADLSim associates each activity with a simple metric called "energy expenditure" quantified in terms of a real number between 0.0 (extremely low expenditure, e.g., during sleep) and 10.0. We also add an additional activity *exercise* with the high energy expenditure 10.0 to enable the simulation of more drastic changes in activity levels upon infection.

3.6.3 Epidemic and disease models

The model consists of two sub-models, the *epidemic model* and the *disease model* (in the remainder of this paper, we refer to these two models collectively as *the epidemic models*). The epidemic model is used to simulate the spread of the disease, i.e., when and where (in which apartment) infections occur. The disease model mimics the various stages of disease development after an individual is infected. The symptoms of the disease are manifested as gradual adjustments of the parameters of the activity models in ADLSim.

Epidemic model: The epidemic model has the three parameters *epidemic start*, *epidemic duration (ED)* and *daily infection probability (DIP)*. The first two determine the first day of the epidemic relative to the be-

ginning of the simulation, and the number of days the epidemic lasts. The last parameter denotes the probability that an apartment is infected for any given day. An infected apartment will stay infected for the total duration of the disease, as determined by the parameters of the disease model. To mimic the phenomenon of post-disease immunity, a given apartment can not be infected more than once during an epidemic. A key advantage of this simple infection model is that the decision of whether an individual apartment is infected in a given day corresponds to a Bernoulli trial. Under this model, the expected number of infected apartments is

$$1 - (1 - DIP)^{ED} \quad (1)$$

Disease model: The disease model has three parameters *symptom onset duration*, *disease duration*, and *symptom recovery duration*. The parameters control the duration of the three main stages of disease development, namely the initial stage of gradual symptom onset, the intermediate stage with stable symptoms, and a final recovery stage during which the symptoms gradually diminish. The duration is specified in terms of the number of days. Symptoms are characterized in terms of their impact on activity model parameters, such as the frequency, duration, energy expenditure, and omission probability of activities. The target values of activity parameters are first configured to reflect a particular disease. Once an individual is infected, the disease model transitions between three states that correspond to the three above mentioned disease stages. The impact of symptoms on parameter values is gradually increased during the *onset* stage and reaches the target values at the end of the period. The parameter values are thereafter kept constant during the *diseased* state, and gradually returned back to their original values during the *recovering* state.

Model output: The effect of the epidemic model is primarily reflected as changes in the activity traces, i.e., in terms of the onset and duration of activities and their energy expenditure. In addition, the model outputs the time at which apartments are infected and when they have recovered.

4 Experiments and Evaluation

This framework has been evaluated for two main factors, the first one is to check if the framework is able to detect an epidemic and the second one is the scalability of the framework. In order to evaluate the detection of an epidemic, ADLSim has been extended to simulate a gastroenteritis epidemic in an apartment complex. In order to evaluate the scalability, a set of experiments

Table 1 Experiment Parametrization

Component	Parameter	Values
ADLSim	Number of apartments	100
	Number of simulated days	90
	Number of runs	3
Symptom Indicators	Threshold for SI event emissions	3 ± 1 , 3 ± 2
	Epidemic model	10, 30, 60
Epidemic model	Daily infection probability (DIP)	0.5%, 1.5%, 3.0%
	Epidemic start	Day 20
	Disease model	3 days
Disease model	Onset duration	3 days
	Disease duration	9 days
	Recovery duration	4 days
<i>Activities affected during disease:</i>		
	Toileting: duration	Increased to 200%
	Toileting: time between visits	Decreased to 30%
	Toileting: probability of interrupting sleep	Increased to 100%
	Breakfast: omission probability	Increased to 75%
	Dinner: omission probability	Increased to 50%
	All activities: activity level	Decreased to 50%

have been conducted to check the scalability of each one of the four main components and the framework as a whole.

4.1 Gastroenteritis epidemic simulation

Gastroenteritis is a quite common disease among the elderly. It can be quickly spread between inhabitants and has been linked to increased mortality among the elderly van Asten and et al. (2011). In France, the elderly were identified as the most frequently infected in gastroenteritis outbreaks between 2006 and 2009 (Barret and et al., 2014). Viral gastroenteritis symptoms may appear within one to three days after infection and may persist as long as ten days at the severe stage. The presented experiments have been designed to simulate the monitoring of *100 apartments* in a nursing complex for *90 days* where a gastroenteritis epidemic may occur. The design of the use-case is based on the literature on gastroenteritis to be realistic, but it is not a medically validated application. Its main aim is to demonstrate the scalability of the proposed semantic ecosystem.

Based on this scenario, it is important that the simulation models are parameterized to yield (1) various degrees of abnormalities in activity during a period of epidemic that approximate what would be expected in realistic scenarios in order to study to which degree these abnormalities are detected by the system (Goal 1) and (2) having a sufficient amount of data to facilitate scalability analysis (Goal 2). To meet Goal 1 and 2, the epidemic model has been parameterized based on real-world studies of outbreaks of gastroenteritis in nursing homes (Gray and et al., 1987; Halvorsrud and Örstavik, 1980; Goller and et al., 2004; Kirk et al., 2010). An overview of the parameters is presented in Table 1. These parameters aim to provide a realistic epidemic simulation but please note that this point is not very crucial for the evaluation since we are focusing on the correctness of the detection of the simulated epi-

demical and not on the validity of the simulated epidemic model.

The epidemic model: The duration and infection rate of gastroenteritis in nursing homes change significantly. For instance, in the three cases in (Gray and et al., 1987), (Halvorsrud and Örstavik, 1980), and (Goller and et al., 2004) the infection rate ranges from 36% to 80%, the number of individuals ranges from 42 to 256, and the time period ranges from 11 to 35 days. Therefore, different parametrizations are used to cover a wide range of scenarios, i.e., with all combinations of ED and DIP where ED is 10, 30, and 60, and DIP is 0.5%, 1.5%, and 3.0%. According to Equation 1, this yields 4.89% to 83.92% infected individuals in the simulations. These infection rates are also reflected in the results in Figure 6 (e) and (f) (described in Section A).

The disease model: Typical symptoms of gastroenteritis are diarrhea, lack of appetite, and fatigue. The effects of diarrhea are simulated as an increase of toileting in three ways: increasing the duration of each instance of the *toileting* activity to 200%, decreasing the duration between the instances to 30%, and during instances, when the need for toileting occurs during nightly sleep, we increase the probability of interrupting sleep to go to the toilet from 0% to 100%. Lack of appetite is simulated by increasing the omission probability of activities *breakfast* and *dinner* from 0% to 75% and 50%, respectively. Finally, fatigue is simulated by reducing the activity level of all activities to 50%. It is also necessary to define the duration of the different stages of the disease. Total disease duration depends very much on the gastroenteritis type (Kirk et al., 2010). We, however, find that a relatively realistic representation of the disease is achieved by setting the onset, disease, and recovery duration to 3, 9, and 4 days respectively.

SI Queries: The accuracy of semantic reasoning relies on the selection of appropriate SI queries to provide reliable input data. SI queries are implemented for all activities that can be affected by the gastroenteritis disease (i.e., SI queries that characterize the total duration and number of instances of *toileting* and activities for having meals). In addition, SI queries are implemented for the total activity level, and for activities that are not affected by the disease (i.e., the number of times an individual had a drink and the total duration of *sleep*) to demonstrate the ability of SI queries to specifically target specific activities. All SI queries have both apartment-specific and complex-wide variants. They are configured to generate events only when the SI level exceeds a pre-defined threshold (depending on the experiments, summarized in Table 1) in addition to providing SI summaries once every day.

4.2 Evaluation of the accuracy of epidemic detection

In order to evaluate the accuracy of the proposed framework, first a set of experiments has been conducted in order to evaluate whether the simulated epidemic and disease models work as expected. The corresponding experiments are presented in the Appendix A. The results of these experiments demonstrate 1) that the epidemic model and all aspects of the SI queries work as intended and 2) that using the most restrictive SI emission thresholds can significantly reduce the number of events emitted and thus save considerable amounts of event processing resources while preserving clearly detectable changes in SI values during the epidemic. Therefore, the following experiment Set 2 relies on SI emission thresholds. This experiment provided a reliable simulation of epidemic so it can be used to evaluate the accuracy of the framework in detecting epidemics.

4.2.1 Experiment Set 2: Evaluation of the framework in detecting a gastroenteritis epidemic

The Experiment Set 2 evaluates the accuracy of the semantic-based detection of simulated diseases and epidemic. For instance, the Epidemic Component (EC) is responsible of the detection process in the proposed framework hence, this experiment set is designed to evaluate the EC and to validate if it detects the infected apartments when an epidemic occurs. For that, 27 sub-experiments have been conducted varying the ED, DIP and the seed values according to Table 1. This experiment allows to compare the results with the ground truth provided by Experiment 1.1, hence, deducing the recall and precision evaluation parameters based on the False/Positives values. The C-SPARQL query used in this experiment is presented in Listing 5. It consists of searching in the last ten days, the apartments where the disease lasts for five days. We chose the number five (5) as a threshold because the gastroenteritis disease becomes worrying when it persists more than three days (with a maximum of 10 days).

Listing 5 C-SPARQL query for the detection of infected apartments for more than three days

```
String queryFindApartments =
"REGISTER QUERY HowManyApartmentEpidemic AS "
+ "PREFIX : <http://www.streamreasoning.org/ontologies/epidemic-
  onto#> "
+ "PREFIX ssn: <http://www.w3.org/ns/ssn/> "
+ "PREFIX sosa: <http://www.w3.org/ns/sosa/> "
+ "PREFIX xsd: <http://www.w3.org/2001/XMLSchema#> "
+ "SELECT DISTINCT ?apart "
+ "?resultTime1 ?resultTime5 "
+ "FROM STREAM <http://streamreasoning.org/streams/diseases>
[RANGE 10d STEP 1s] "
+ "WHERE { "
+ "??diseaseObservation1 sosa:hasFeatureOfInterest ?apart ."
+ "??diseaseObservation2 sosa:hasFeatureOfInterest ?apart ."
+ "??diseaseObservation3 sosa:hasFeatureOfInterest ?apart ."
+ "??diseaseObservation4 sosa:hasFeatureOfInterest ?apart ."
+ "??diseaseObservation5 sosa:hasFeatureOfInterest ?apart ."
```

```
+ "??diseaseObservation1 sosa:resultTime ?resultTime1 ."
+ "??diseaseObservation2 sosa:resultTime ?resultTime2 ."
+ "??diseaseObservation3 sosa:resultTime ?resultTime3 ."
+ "??diseaseObservation4 sosa:resultTime ?resultTime4 ."
+ "??diseaseObservation5 sosa:resultTime ?resultTime5 ."

+ "??diseaseObservation1 sosa:hasSimpleResult ?disease ."
+ "??diseaseObservation2 sosa:hasSimpleResult ?disease ."
+ "??diseaseObservation3 sosa:hasSimpleResult ?disease ."
+ "??diseaseObservation4 sosa:hasSimpleResult ?disease ."
+ "??diseaseObservation5 sosa:hasSimpleResult ?disease ."

+ "FILTER (?resultTime5 = ?resultTime4 + (1000*60*60*24)) ."
+ " && (?resultTime4 = ?resultTime3 + (1000*60*60*24)) ."
+ " && (?resultTime3 = ?resultTime2 + (1000*60*60*24)) ."
+ " && (?resultTime2 = ?resultTime1 + (1000*60*60*24)) ."
+ " && (?resultTime1 > 1465200000 && ?resultTime5 < 4320000000) ."
+ "};"
```

The evaluation of Experiment 2 relies on the F1-score presented as F1 formula in (2).

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (2)$$

This F1-score is equal to the weighted average of Precision and Recall functions. The Precision function is equal to the ratio of correctly predicted positive apartments to the total predicted positive apartments.

$$Precision = \frac{TP}{TP + FP} \quad (3)$$

The Recall function is equal to the ratio of correctly predicted positive apartments to the all apartments in the actual monitoring system. Therefore, the F1-score takes True Positives (TP), False Positives (FP) and False Negatives (FN) into account. True Positives count infected apartments that are detected during the infection period. False positive count apartments detected as infected outside of the epidemic period. False Negatives count infected apartments that are not detected as infected during the epidemic period.

$$Recall = \frac{TP}{TP + FN} \quad (4)$$

4.2.2 Results and Discussion

The results of the nine sub-experiments with ED=10 of the Experiment Set 2 are presented in Table 2. Table 3 presents an excerpt of the 18 sub-experiments (ED=30 and ED=60) as they show the same behavior.

Table 2 shows that when the ED is equal to ten days, the F1-score varies between 0.846169 and 0.9735. Precision is at its lowest when the DIP is at its highest with a value of 0.898. Recall evolves with the same pattern and has its lowest value (0.846169) with a DIP of 3%. Table 3 presents the same behavior with a small diminution of the F1-score when the ED increases (F1-score: 0.8-0.94 for ED=30 and 0.79-0.93 for ED=60).

Trends in the experiment results show that the F1-score is diminishing when the DIP increases. This can be explained by the increasing number of events to deal

with in the C-SPARQL module. Indeed, C-SPARQL is known to have a rather bad resolution (1s) (Barbieri and et al., 2010) which means that it is not fully able to process large amount of data in a short time period. So, the more events to process, the lower the F1-score. It should be noted that the number of infected apartments increases when the duration of the DIP is increased leading to an increasing number of symptom indicators to process. The DIP has then a direct impact on both precision and recall. The false negatives for the lowest DIP has been studied also to understand why the precision is not at 100%. This result makes sense as we are looking for five consecutive days of infection in the C-SPARQL query, but the framework does not use a stochastic model to generate infection on ADLSim whose traces serves as a ground truth for the computation of the F1-score. Therefore, it can happen that infection are generated with a day without symptoms during a five-days time window which leads to no detection by the C-SPARQL module.

4.3 Evaluation of the scalability of the proposed architecture

A key contribution in this work is to enable semantic data analysis for large-scale scenarios. Experiment Set 3 have been conducted to evaluate the scalability over semantic reasoning. It consists of six experiments that evaluate scalability aspects of each of the five components in the system individually.

4.3.1 Experiment Set 3: Scalability Analysis

Experiment Set 3 is designed to analyse whether each component in the proposed architecture is scalable, i.e., it has low processing overhead that scales (sub-)linearly with input workload, and that the architecture as a whole facilitates semantic analysis in scenarios with high volume and velocity data.

Thus, this experiment set measures the processing overhead of each individual component and relates this to the number of input events processed and output events generated. For the architecture, it is important to show that the event frequency is significantly reduced on the path from the low-level CEP pre-processing with SI queries, which are highly efficient but lack semantic reasoning, to the high-level semantic reasoning which naturally requires more processing time per event.

The main goals of this experiment are to measure (1) the time spent for event and query processing, and (2) the number of output events produced. For each experiment, one additional downstream component is enabled to study the added processing overhead of each new component.

The six experiments have been conducted according to the following protocol: Experiment 3.1 targets ADLSim and the epidemic models. In Experiment 3.2 and 3.3, SI queries are enabled. In Experiment 3.4, semantic reasoning is enabled but the events produced are not transmitted over MQTT since we focus on the overhead of semantic enrichment. In Experiment 3.5, the DC is enabled. As the framework runs a SPARQL query only when new sets of symptoms are detected by the DC, this experiment measures the processing time of each SPARQL query. In Experiment 3.6, all components in the architecture are enabled. This experiment measures the processing time of each C-SPARQL query running when new triples enter the window. In all experiments, except Experiment 3.3, the SI emission thresholds are highly restrictive "high" and "low". In Experiment 3.3, the thresholds from Experiment 1.2 are used, i.e., to generate SI values whenever the SI value changes from three ("normal") to "very high" and "very low".

4.3.2 Results and Discussion

The results from Experiments 3.1 to 3.4 are presented in Table 4, and those from Experiments 3.5 and 3.6 are presented in Table 5. Each row has results for one of the six experiments. The first column denotes the experiment and the involved components. The second and third columns show the number of input events processed by the component, and the number of output events produced as a result, respectively. Since the output events from one component constitute the input events of the component that subsequently processes these events, the same number of input and output events is detected. Since ADLSim generates first event stream, the "number of input events" is not applicable (N/A). The fourth column presents the processing duration spent per workload. Both the numbers of events and processing duration depend on the parametrization and random variables of the particular sub-experiments. Therefore, each experiment presents the ranges of event numbers and processing times observed across all 27 sub-experiments.

As a result, ADLSim produces between 251386 and 256898 events, where each event denotes the beginning and end of activity instances. About one microsecond is spent generating each output event, which is at least one order of magnitude lower than the processing per workload unit spent in any other component. We also see that ADLSim produces at least two orders of magnitude more events than any other component. For semantic reasoning to be feasible, it is therefore crucial that the SI queries significantly reduce this number of events. Experiments 3.2 and 3.3 confirm that the num-

Table 2 F1-score Table for nine sub-experiments of Experiment 2.1, ED=10

ED	10								
DIP	0.5%			1.5%			3%		
Seed	1	2	3	1	2	3	1	2	3
Precision	0.9735	0.963299	0.933	0.9555	0.9499	0.9163	0.9069	0.89855	0.898
Recall	0.9735	0.941875	0.9688	0.90299	0.896	0.8853	0.8075	0.81422	0.799994
F1-score	0.9735	0.95247	0.950563	0.922163	0.900533	0.854318	0.854309	0.854309	0.846169

Table 3 F1-score Table for nine sub-experiments of Experiment 2.1, ED=60

ED	60								
DIP	0.5%			1.5%			3%		
Seed	1	2	3	1	2	3	1	2	3
Precision	0.909444	0.9322	0.9533	0.89185	0.896	0.906	0.929	0.9034	0.902
Recall	0.71819	0.8233	0.9088	0.74765	0.723	0.7166	0.753	0.7774	0.71
F1-score	0.80258	0.8743723	0.9305183	0.813409	0.800257	0.800246	0.831792	0.835677	0.794566

Table 4 Results from the scalability evaluation for Experiments 3.1, 3.2, 3.3 and 3.4.

Experiment	Number of input events	Number of output events	Processing time
Experiment 3.1: ADLSim and epidemic models	N/A	251386 to 256898	0.96 to 1.02 μ s / output event
Experiment 3.2: CEP with SI threshold <i>low</i> and <i>high</i>	251386 to 256898	25788 to 28613	31.05 to 36.85 μ s / input event
Experiment 3.3: CEP with SI threshold <i>very low</i> and <i>very high</i>	251386 to 256898	2367 to 3851	30.70 to 35.14 μ s / input event
Experiment 3.4: Semantic enrichment with SI threshold <i>very low</i> and <i>very high</i>	2357 to 3851 (equivalent to 16499 to 26957 observations)	16499 to 26957	1.90 to 4.61 to ms / input event

ber of events is reduced by two and three orders of magnitude, respectively. By comparing the difference between the processing time of ADLSim and that of the SI queries, it is clear that the processing overhead of SI queries per input event is less than 37 microseconds in all cases. With an average of 28 activity instances per day per apartment, the SI queries are able to process three months of activity traces within 15 seconds, which is clearly more than sufficient to facilitate rapid detection of diseases and epidemics. Furthermore, when comparing the results from Experiments 3.2 and 3.3., it shows that the number of output events increases by one order of magnitude with less restrictive SI emission thresholds, but that the added processing duration increases only slightly. This shows that the processing overhead of SI queries primarily depends on the number of input events.

The SI component is receiving input from ADLSim with "*very high*" and "*very low*" thresholds. Each input consists of an event composed of seven SI per apartment per day. In Experiment 3.4, the SI component semantically enrich each SI as a separate observation, hence the same input/output is observed. However, the difference shown in Table 4 can be explained by multiplying the number of input events by seven. Regarding the semantic overhead, Experiment 3.4 shows that the processing overhead for each SI input is between 1.90 and 4.61 ms. The output of the SI component is the input for the Dis-

ease Component that applies SPARQL queries to infer a disease. This should decrease the number of events in the output by aggregating the input data.

The results of Experiment 3.5 presented in Table 5 confirm the decreasing number of events in the output. Moreover, the processing overhead is between 46 to 1000 ms per query which is the time to search in the DO ontology and extract the result. Each SPARQL query depends on the number of symptoms received and the detected diseases. This experiment shows that the processing overhead also depends on the input data. Finally, in Experiment 3.6 the C-SPARQL component (1) receives the output events from the SPARQL component, (2) applies queries with temporal considerations and (3) generates 0 to 60 messages related to the number of infected apartments. This experiment shows that the overall system is able to substantially decrease the number of events. The first component in ADLSim sends a number of events in the range of 251386-256898 while the last component (DC) generates a number of events in the range of 0-60. The processing overhead of C-SPARQL is between 5 to 10 ms depending on the amount of RDF streams received and processed per time unit.

4.4 Summary and key insights

Results previously presented demonstrate that the proposed architecture with the semantic reasoning is suit-

Table 5 Results from the scalability evaluation for experiments 3.5 and 3.6

Experiment	Number of input events	Number of output events	Processing time
Experiment 3.5: SPARQL	16499 to 26957	2205 to 3086	46-1000 ms/query
Experiment 3.6: C-SPARQL	2205 to 3086	0-60	5-10ms/query

able for diseases and epidemic detection with a F1-score above 0.79. The precision of the detection of apartments where there is an infection is around 90%. Regarding the epidemic detection, the proposed system is able to detect an outbreak of a particular disease with three to five days deviation. The SI queries are able to process large-scale activity data with very low processing overhead, even on a regular consumer laptop. Thus, SI queries are by far efficient enough to facilitate rapid detection of diseases and epidemics in a large scale, on-line analysis setting. The semantic reasoning techniques SPARQL and C-SPARQL have efficient knowledge-based processing with an affordable overhead processing as in just a few seconds we complete querying of three months of data. The results also show that the processing time per unit of workload per component is similar across all parameterizations in all experiments. This means that the total processing overhead increases linearly with the the number of apartments and/or the number of activities performed per day. These results demonstrate that the proposed framework is scalable. These results demonstrate that the proposed ontology-based solution with semantic reasoning can be used for the detection of epidemic outbreaks and it is sufficiently scalable.

5 Conclusions and Future work

In this paper, a scalable semantic framework is proposed for disease and epidemic monitoring in a compound of elderly. This framework would have significant effect in improving safety and aging in place for elderly as long as they desire staying at home. For example, this framework could be connected to alarm systems of kitchen, compound managers and authorities so they react fast when a problem issues. It also can be helpful for existing healthcare monitoring application such as fall detection systems when the the framework detects anomalies in ADLs. This framework promotes scalability without sacrificing the fundamental ability of semantic reasoning to detect adverse events of the type found during epidemics. This framework solves this problem by preceding semantic reasoning with CEP processing that reduces the large number of ADL events to a stream of SI with at least one order of magnitude fewer events. The resulting SI are analyzed according to the DO ontology using SPARQL to detect

individual disease instances, which is used as input to C-SPARQL for continuous detection of epidemics. To enable realistic evaluation, ADLSim has been extended and added to the framework as the first element in the pipeline to simulate realistic ADL sequences, and to simulate the impact of diseases on ADL.

While the queries and model parameterizations are inspired by real-world instances of gastroenteritis, the current framework focuses on the accuracy of how our queries can detect what was simulated. Future work will focus on the accuracy of the queries and models, therefore will include working with health professionals to develop and evaluate models for more realistic epidemic simulation. Moreover, RDF-stream query languages are a trending and new research topic. It would be interesting to test other RDF stream processors such as CQELS (Le-Phuoc and et al., 2011) for instance to deal with the granularity issue that have been encountered with C-SPARQL.

Acknowledgements This work has been funded by the COST Action IC 1303 AAPELE; the Research Council of Norway through the CESAR project (Grant nb. 250239/O70); and the project IoTSec from the french Region Limousin

References

- Abdallah ZS, et al (2018) Activity recognition with evolving data streams: A review. *ACM Computing Surveys (CSUR)* 51(4):71
- Alaya MB, et al (2014) OM2M: Extensible ETSI-compliant M2M Service Platform with Self-configuration Capability. *Procedia Computer Science*
- Alessandra M, et al (2016) Recognition of Daily Gestures with Wearable Inertial Rings and Bracelets. *Sensors* 16(8):1341
- Aran O, et al (2016) Anomaly detection in elderly daily behavior in ambient sensing environments. In: *International Workshop on Human Behavior Understanding*, Springer, pp 51–67
- van Asten L, et al (2011) Unspecified gastroenteritis illness and deaths in the elderly associated with norovirus epidemics. *Epidemiology*
- Avci A, et al (2010) Activity recognition using inertial sensing for healthcare, wellbeing and sports applications: A survey. In: *Architecture of computing*

- systems (ARCS), 23rd international conference, pp 1–10
- Barbieri DF, et al (2010) C-sparql: a continuous query language for rdf data streams. *International Journal of Semantic Computing* 4(01):3–25
- Barret A, et al (2014) Surveillance for outbreaks of gastroenteritis in elderly long-term care facilities in france, november 2010 to may 2012. *Eurosurveillance* 19(29)
- Bispo KA, Rosa NS, Cunha PR (2015) Sitrus: Semantic infrastructure for wireless sensor networks. *Sensors* 15(11)
- Curry E (2004) Message-oriented middleware. *Middleware for communications* pp 1–28
- Donnelly K (2006) Snomed-ct: The advanced terminology and coding system for ehealth. *Studies in health technology and informatics* 121:279
- Eckert M, et al (2011) A cep babelfish: Languages for complex event processing and querying surveyed. In: *Reasoning in Event-Based Distributed Systems*, Springer, pp 47–70
- espertech (2006) Esper cep engine. URL <http://www.espertech.com/>, accessed: 16-May-2019
- Fernández-Llatas C, et al (2011) Behaviour patterns detection for persuasive design in nursing homes to help dementia patients. In: *Engineering in medicine and biology society, EMBC, 2011 annual international conference of the IEEE, IEEE*, pp 6413–6417
- Ghayvat H, et al (2018) Smart home based ambient assisted living: Recognition of anomaly in the activity of daily living for an elderly living alone. In: *2018 IEEE (I2MTC), IEEE*, pp 1–5
- Goller J, et al (2004) Long-term features of norovirus gastroenteritis in the elderly. *Journal of Hospital Infection* 58(4):286–291
- Gray J, et al (1987) An outbreak of gastroenteritis in a home for the elderly associated with astrovirus type 1 and human calicivirus. *Journal of medical virology* 23(4)
- Haller A, et al (2018) The modular ssn ontology: A joint w3c and ogc standard specifying the semantics of sensors, observations, sampling, and actuation. *Semantic Web* 10(01):09–32
- Halvorsrud J, Örstavik I (1980) An epidemic of rotavirus-associated gastroenteritis in a nursing home for the elderly. *Scandinavian journal of infectious diseases* 12(3):161–164
- Hochgatterer A, et al (2011) Requirements for a behaviour pattern based assistant for early detection and management of neurodegenerative diseases. In: *PervasiveHealth, 2011 5th International Conference on, IEEE*, pp 346–353
- Hoque E, Dickerson RF, Preum SM, Hanson M, Barth A, Stankovic JA (2015) Holmes: A comprehensive anomaly detection system for daily in-home activities. In: *International Conference on Distributed Computing in Sensor Systems (DCOSS), IEEE*
- Hsu HH, Chen CC (2010) Rfid-based human behavior modeling and anomaly detection for elderly care. *Mobile Information Systems* 6(4):341–354
- Katz S, et al (1970) Progress in development of the index of adl. *The gerontologist* 10:20–30
- Khrouf H, et al (2016) Waves: big data platform for real-time rdf stream processing. In: *SR workshop at ISWC*
- Kibbe WA, et al (2014) Disease ontology 2015 update: an expanded and updated database of human diseases for linking biomedical knowledge through disease data. *Nucleic acids research* 43:D1071–D1078
- Kim J, Lee JW (2014) OpenIoT: An open service framework for the Internet of Things. In: *2014 IEEE World Forum on Internet of Things (WF-IoT), IEEE*
- Kirk MD, Veitch MG, Hall GV (2010) Gastroenteritis and food-borne disease in elderly people living in long-term care. *Clinical Infectious Diseases* 50(3):397–404
- Krishnan NC, Cook DJ (2014) Activity recognition on streaming sensor data. *Pervasive and Mobile Computing* 10:138–154
- Kristiansen S, et al (2016) Smooth and crispy: integrating continuous event proximity calculation and discrete event detection. In: *Proceedings of the 10th ACM International Conference on Distributed and Event-based Systems, ACM*, pp 153–160
- Kristiansen S, et al (2018) An activity rule based approach to simulate adl sequences. *IEEE Access* 6:12551–12572
- Kwok T, et al (2002) Parallel fuzzy c-means clustering for large data sets. In: *European Conference on Parallel Processing, Springer*, pp 365–374
- Lampoltshammer aa Thomas J (2014) Use of local intelligence to reduce energy consumption of wireless sensor nodes in elderly health monitoring systems. *Sensors* 14(3):4932–4947
- Le-Phuoc D, et al (2011) A native and adaptive approach for unified processing of linked streams and linked data. In: *International Semantic Web Conference, Springer*, pp 370–388
- Lohr C, Tanguy P, Kerdreux J (2015) xaal: A distributed infrastructure for heterogeneous ambient devices. *Journal of intelligent systems* 24(3):321–331
- Lotfi A, et al (2012) Smart homes for the elderly dementia sufferers: identification and prediction of abnormal behaviour. *Journal of ambient intelligence and humanized computing* 3(3):205–218

- Margara A, Cugola G, Collavini D, Dell’Aglia D (2018) Efficient temporal reasoning on streams of events with *dotr*. In: Gangemi A, Navigli R, Vidal ME, Hitzler P, Troncy R, Hollink L, Tordai A, Alam M (eds) *The Semantic Web*, Springer International Publishing, pp 384–399
- Meditskos G, Dasiopoulou S, Kompatsiaris I (2016) *MetaQ: A knowledge-driven framework for context-aware activity recognition combining SPARQL and OWL 2 activity patterns*. *Pervasive and Mobile Computing* DOI 10.1016/j.pmcj.2015.01.007
- Meng L, Miao C, Leung C (2017) Towards online and personalized daily activity recognition, habit modeling, and anomaly detection for the solitary elderly through unobtrusive sensing. *Multimedia Tools and Applications*
- Mohammed O, et al (2012) Building a diseases symptoms ontology for medical diagnosis: an integrative approach. In: *Future Generation Communication Technology (FGCT), 2012 International Conference on*, IEEE, pp 104–108
- Noury N, et al (2016) Building a spatial-temporal index to detect the global pattern deviations in daily activities of elderly subjects. In: *18th International Conference on e-Health Networking, Applications and Services*, IEEE
- Novák M, et al (2012) Unobtrusive anomaly detection in presence of elderly in a smart-home environment. In: *ELEKTRO, 2012*, IEEE, pp 341–344
- Onofri L, et al (2016) A survey on using domain and contextual knowledge for human activity recognition in video streams. *Expert Systems with Applications*
- Perriot B, et al (2014) Characterization of physical activity in copd patients: Validation of a robust algorithm for actigraphic measurements in living situations. *IEEE journal of biomedical and health informatics* 18(4):1225–1231
- Pierleoni P, et al (2015) A high reliability wearable device for elderly fall detection. *IEEE Sensors Journal* 15(8):4544–4553
- Pires I, et al (2016) From data acquisition to data fusion: a comprehensive review and a roadmap for the identification of activities of daily living using mobile devices. *Sensors* 16(2):184
- Quilitz B, Leser U (2008) Querying distributed rdf data sources with sparql. In: *European semantic web conference*, Springer, pp 524–538
- Razzaque MA, et al (2016) Middleware for Internet of Things: A Survey. *IEEE Internet of Things Journal* 3(1):70–95
- Rodríguez-Molina J, et al (2013) SMARc: a proposal for a smart, semantic middleware architecture focused on Smart City energy management. *International Journal of Distributed Sensor Networks*
- Ryoo MS, Aggarwal JK (2006) Recognition of composite human activities through context-free grammar based representation. In: *Computer vision and pattern recognition, IEEE computer society conference, IEEE*, vol 2, pp 1709–1718
- Sareen S, et al (2017) Secure internet of things-based cloud framework to control zika virus outbreak. *International journal of technology assessment in health care* 33(1):11–18
- Sareen S, Sood SK, Gupta SK (2016) Iot-based cloud framework to control ebola virus outbreak. *Journal of Ambient Intelligence and Humanized Computing* pp 1–18
- Schriml LM (2018) Symptom ontology. URL <https://www.ebi.ac.uk/ols/ontologies/symp>, accessed: 26-February-2019
- Schriml LM, et al (2011) Disease ontology: a backbone for disease semantic integration. *Nucleic acids research* 40(D1):D940–D946
- Serna A, et al (2007) Modeling the progression of alzheimer’s disease for cognitive assistance in smart homes. *User Modeling and User-Adapted Interaction* 17(4):415–438
- Strausbaugh LJ, et al (2003) Infectious disease outbreaks in nursing homes: an unappreciated hazard for frail elderly persons. *Clinical infectious diseases* 36(7):870–876
- Thakar AT, Pandya S (2017) Survey of iot enables healthcare devices. In: *Computing Methodologies and Communication (ICCMC), 2017 International Conference on*, IEEE
- Ukil A, et al (2016) Iot healthcare analytics: The importance of anomaly detection. In: *Advanced Information Networking and Applications (AINA), 2016 IEEE 30th International Conference on*, IEEE, pp 994–997
- Van Kasteren T, et al (2008) Accurate activity recognition in a home setting. In: *Proceedings of the 10th international conference on Ubiquitous computing*, ACM, pp 1–9
- Varatharajan R, et al (2017) Wearable sensor devices for early detection of alzheimer disease using dynamic time warping algorithm. *Cluster Computing* pp 1–10
- Wolf P, et al (2010) openaal-the open source middleware for ambient-assisted living (aal). In: *AALIANCE conference*, Malaga, Spain, pp 1–5
- Wu J, et al (2018) Sensor fusion for recognition of activities of daily living. *Sensors* 18(11):4029
- Yu M, et al (2012) A posture recognition-based fall detection system for monitoring an elderly person in a smart home environment. *IEEE transactions on information technology in biomedicine* 16(6):1274–1286

- Zgheib R, et al (2017) A flexible architecture for cognitive sensing of activities in ambient assisted living. In: *Enabling Technologies: WETICE, IEEE 26th International Conference, IEEE*, pp 284–289
- Zgheib R, et al (2019) Semantic middleware architectures for iot healthcare applications. In: *Enhanced Living Environments, Springer*, pp 263–294
- Ziaefard M, Bergevin R (2015) Semantic human activity recognition: a literature review. *Pattern Recognition* 48(8):2329–2345
- Ziakas PD, et al (2016) Prevalence and impact of clostridium difficile infection in elderly residents of long-term care facilities, 2011: A nationwide study. *Medicine* 95(31)

A Evaluation of epidemic simulation in ADLSim

To obtain reliable results from Experiment Sets 2 and 3, it is necessary to first ensure that our simulation models and SI queries work as intended. This section presents a set of experiments designed for this purpose, and discusses the results.

A.0.1 Experiment Goals and Design

The main goals of these experiments are to show that (1) the epidemic model behaves as intended, (2) the output from the SI queries correspond to the impact of the epidemic model, and (3) that the results vary realistically with the values of key parameters (listed in Table 1).

We conduct two experiments, Experiment 1.1 and 1.2. Each experiment includes one sub-experiment per combinations of the three ED and three DIP parameters shown in Table 1. This enables us to study epidemics ranging from very mild to very severe. We conduct each sub-experiment three times with different seeds and which results in $3 \times 3 \times 3 = 27$ sub-experiments per experiment. In Experiment 1.1, we use highly conservative thresholds for SI event emissions. SI values are emitted only when they change to *“very low”* or *“very high”*. In Experiment 1.2, we increase the sensitivity of the queries to emit events whenever the value changes from *“normal”* to any other value. This way we can study the impact on the number of events emitted for further semantic reasoning. In addition, we generate every simulated day 24-hour SI statistics regarding the entire apartment complex.

A.0.2 Results and Discussions

To get a comprehensive understanding of our results, we present statistics across all runs and parameter values as well as plots that provide a deeper look at individual runs for a sub-set of the results.

Figure 6 presents the results from Experiment Set 1. Graphs (a), (b), (c), and (d) present results from the first run in Experiment 1.1 with ED = 30 and DIP = 1.5% (left) and 3% (right). The epidemic begins at simulated Day 20. Graphs (a) and (b) show the complex-wide 24-hour SI statistics (y-axes) for all simulated days (x-axes), and Graphs (c) and (d) show the number of apartments (y-axes) with SI values *“very low”* and *“very high”* for all simulated days (x-axes). Graphs (e) and (f) show statistics across all runs for all parameter values from

Experiment 1.1 and 1.2. They show the average (lines) and SD (error bars) of the number of emitted SI events (left y-axes) across all three runs per combination of DIP (x-axes) and ED (with different line colors). Note the differences in the left y-axes in Graphs (e) and (f). The box plots show the average number of the infected individual across all three runs (right y-axes) per combination of DIP (x-axes) and ED (with different box colors).

The results in Graphs (a), (b), (c), and (d) show a clear impact on the complex-wide SI of the simulated epidemic, namely that the SI values during the epidemic have significantly larger absolute values during the epidemic as well as before and after the epidemic. This effect is also clearly larger with DIP = 3% than with DIP = 1.5%. We also see that only the queries affected by the disease are impacted, i.e., the bottom five queries. Notice that although the epidemic lasts from Day 20 to Day 50, the main impact on the complex-wide SI values is found somewhat later. Three factors contribute to this. The most important factor is that a certain amount of time after the epidemic onset is required for a sufficient number of individuals to be infected to have a significant impact on the aggregate activity magnitudes in the whole apartment complex. Second, due to the gradual onset of symptoms, three days pass after an individual is infected until the full effect of the symptoms is manifested. Finally, at the end of the epidemic period, a certain number of individuals are still symptomatic for some time until recovery, which shows up in the SI queries. There are occasional, abnormal SI values before and after the epidemic explained by natural variations in behavior. These are trivially exposed as false positives by jointly considering a larger set of samples and/or samples from multiple SI queries. These results thus indicate that the complex-wide SI queries respond correctly to the simulated epidemic.

The results in Graphs (c) and (d) show that also the apartment-specific SI values are as expected, i.e., the period with the epidemic results in a significantly larger number of apartments with indications of decreased activity levels reduced intake of food, and increased toileting. The effect is again clearly more pronounced with the largest DIP. We can thus conclude that also the apartment-specific SI work as intended.

Figure 6 (e) and Figure 6 (f) show that the least restrictive SI thresholds (*“high”* and *“low”*, Figure 6 (e)) yield an order of magnitude more SI events than the most restrictive thresholds (*“very high”* and *“very low”*, Figure 6 (f)). This has a significant impact on the number of resources required for subsequent analysis. We also see that the epidemic is clearly detectable by visual inspection in Graphs (a) and (b) even with the most restrictive thresholds. It is therefore advisable in this scenario to employ highly restrictive SI thresholds to minimize the emission rate of SI events. The box plots finally show that the number of infected apartments agrees well with Equation 1, and that these numbers are proportional to the number of emitted SI. Thus, we find that the SI thresholds and the epidemic models work as intended.

Summary and key insights: Results from Experiment Set 1 demonstrate that the epidemic model and all aspects of the SI queries work as intended. We also find that using the most restrictive SI emission thresholds can significantly reduce the number of events emitted and thus save considerable amounts of event processing resources while preserving clearly detectable changes in SI values during the epidemic. We therefore decide to use these SI emission thresholds in Experiment Set 2.

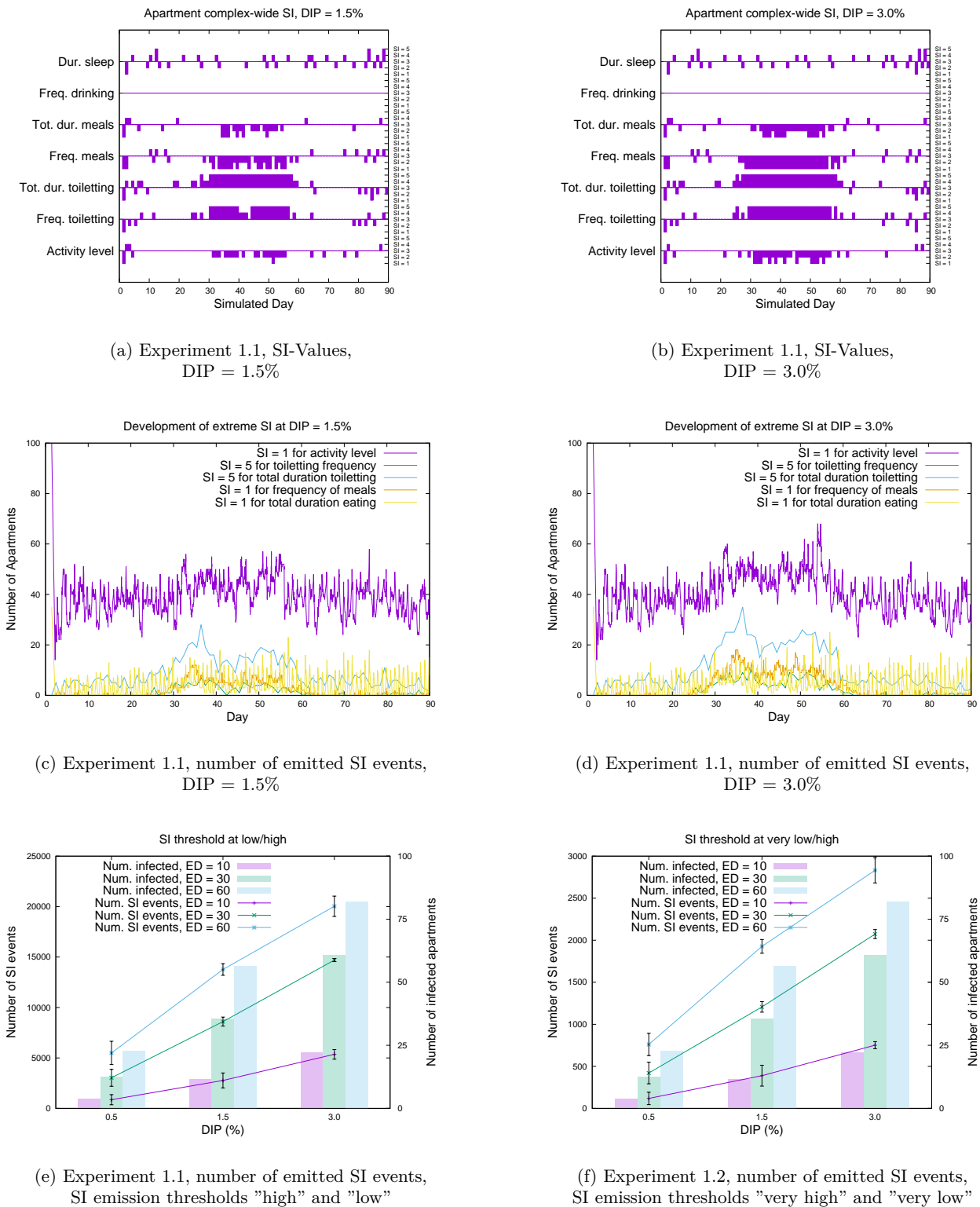


Fig. 6 Results from Experiments Set 1.