



# Towards ambient assisted cities using linked data and data analysis

Rubén Mulero<sup>1</sup> · Vladimir Urosevic<sup>2</sup> · Aitor Almeida<sup>1</sup> · Christos Tatsiopoulos<sup>2</sup>

Received: 15 November 2017 / Accepted: 13 June 2018 / Published online: 28 June 2018  
© The Author(s) 2018

## Abstract

As citizens' age increases, smart cities must adapt to help them to age properly. The objective of the City4Age project is to create the future ambient assisted cities that will help the citizens to deal with mild cognitive impairments (MCI) and frailty. In this paper we present two of the tools developed during the project. The first one is a city-wide context-manager, which allows to store the citizens information using a semantic representation and share it following the linked open data paradigm. The second one are the individual care monitoring dashboards, which use the stored information to help the caregivers to analyze and interpret the citizens' behaviour, allowing to detect risks related to MCI and frailty.

**Keywords** Smart cities · Ambient assisted living · Linked open data · Data analysis

## 1 Introduction

As a result of the growth of the urban population worldwide (Nations 2014) cities are consolidating their positions as one of the central elements in human organization. The concentration of resources and structures around cities offer new opportunities. *Smart Cities* (Caragliu et al. 2011; Shapiro 2006) are emerging as a paradigm to take advantage of these opportunities to improve the lives of their citizens. The City4Age<sup>1</sup> platform is a H2020 research and innovation project with the aim of enabling age-friendly cities. The project aims to create an innovative framework of ICT tools and services that can be deployed by European (and global) cities to improve the lives of senior citizens. To do this City4Age focuses specifically to citizens suffering from early stage frailty and Mild Cognitive Impairments (MCI). In order to help the users of the urban geriatric care and the

City4Age system, the project has two objectives: to enhance the early detection of the risks and to provide personalized interventions that can help the elderly population to improve their daily life by promoting positive behavior changes. To be able to do this, the resulting system must tackle several challenges: to be able to integrate the heterogeneous data gathered from several cities, to support different granularity levels in the data integration of each city, to manage data from a large number of citizens in each city by ensuring its scalability and to provide an unified vision of the data to allow third parties to securely access it in order to boost the research in early detection of frailty and MCI.

To tackle these problems we have created a Smart Cities middleware layer, composed from two main elements: a city-wide context data manager and the analytical dashboards. These two components allow to gather, integrate, manage, analyze, and contextualize/interpret the frailty and MCI related data captured by the Smart City. The city-wide context-data management system serves as a centralized information repository for the project. In the past, we have worked on developing platforms to manage the context (Dey 2001) in intelligent environments, (Almeida and López-de-Ipiña 2012a; Vazquez et al. 2009; Almeida et al. 2009) but limiting ourselves to a single home or building. In the case of City4Age, the created context-manager is able to simultaneously manage and integrate the information produced by several cities, while being flexible enough to allow each city to decide the level of abstraction of the data that they

✉ Rubén Mulero  
ruben.mulero@deusto.es

Vladimir Urosevic  
vladimir.urosevic@belit.co.rs

Aitor Almeida  
aitor.almeida@deusto.es

Christos Tatsiopoulos  
christos@belit.co.rs

<sup>1</sup> DeustoTech Institute of Technology, University of Deusto, Bilbao, 48007 Vizcaya, Spain

<sup>2</sup> Belit Ltd., Trg Nikole Pasica 9, 11000 Belgrade, Serbia

<sup>1</sup> <http://www.city4ageproject.eu/>.

provide. The contributions of the City4Age context-manager are threefold. The first one is to provide a data repository for all the information generated in the cities (i.e. activities of daily living, behavioural patterns, detected MCI and frailty risks, proposed interventions, etc.), integrating it and allowing for the analytic algorithms that use that data to access it in a fast and timely manner. The second one is to provide a mechanism to give semantic meaning to the stored information and to share it with third parties while preserving the user privacy. To do this we have designed an architecture with two main elements: (1) A high performance REST application programming interface (API) that allows to manage large quantities of data easily and (2) a Linked Open Data (LOD) API that maps the information in the database to Ontology Web Language (OWL) (Hitzler et al. 2009) providing semantic meaning to the stored data and making it easier to share. Following the Linked Open Data paradigm (Bizer et al. 2009), we ensure that the provided data will be easily understandable and usable by third-parties, being them humans or machines. The third one are the analytical dashboards for caregivers and geriatricians that are integrated with the context-data manager. The data integrated by the context-data manager needs to be presented and visualized in a manner that will be understandable by the medical experts. The Individual Care Monitoring Dashboards allow the interactive visualization of the users' behavioural data, focused on the representation of a single user's aggregated data. They also serve as a collaborative environment for the experts to assess/annotate the data in order to quantify and interpret detected risks related to Mild Cognitive Impairments and frailty.

This paper has the following structure: Section 2 reviews previous work related to context management and *smart cities*, presenting the contributions of our system. Section 3 introduces the architecture and characteristics of the City4Age city-wide context manager. Section 4 describes the City4Age data model for Ambient Assisted Cities. Section 5 presents the analytical dashboards for user behavior analysis. Section 6 describes the evaluation of the system, presenting the performed experiments and discussing the deployment of the system in six cities worldwide. Finally, Sect. 7 draws some conclusions and outlines future work.

## 2 State of the art

### 2.1 Smart cities

In the last years, researchers have combined different technologies to increase the intelligence of the services provided by cities to citizens. The adoption of the Internet of Things (IoT) and the citizens' active inclusion in the cities' decision making processes are creating new research opportunities.

New requirements are arising from the citizens related to their daily living. With these requirements in mind, *smart cities* appear (Komninos 2002; Torres et al. 2005; Komninos 2009) to help and support citizens in different areas such as health, administration, energy or transportation.

*Smart cities* are the natural evolution of our actual society, not only because increasingly more humans are living in urban environments, but also because the interdependence of the citizens with the city is stronger than ever, and in some areas, this can help to improve the existing infrastructures to satisfy the citizens demands (Alawadhi et al. 2012). The adoption of Internet of Things (IoT) paradigm and the creation of the network infrastructures to connect smartphones, smartbands, sensors, cameras and all new technologies connected via the Internet, are opening new capabilities to connect humans directly with their cities and their environments (Theodoridis et al. 2013). In addition, it creates new venues to gather users' data and analyse them to obtain useful information and extract behavioural patterns.

There are several relevant developed or deployed middleware components and architectures for smart environments that manage both the sensory information and the actions that are executed to react to those events. In Ji et al. (2014), authors present a cloud-based car parking middleware that helps the students of a university to find the optimal parking lot. Authors in Krylovskiy et al. (2015) present an implementation of microservices architecture to build smart cities using an IoT platform with the aim of increasing the energy efficiency of a city in a district level. In Azzara et al. (2013) the authors make an overview of the middleware solutions targeted in Wireless Sensor Networks which allows to interconnect large scale applications with programs that needs to manage each different IoT device.

### 2.2 Middleware technologies for smart cities

In the recent years, the paradigm of Linked Open Data is paving a new way for gathering and sharing knowledge between computers, and has enabled the creation of different services (Ojo et al. 2015). The cities play a pivotal role to gather data directly from the citizens. Hence, a new research area has been opened and various different projects have been launched to study the impact of the *smart cities* in the citizens' lives while they gather data to be used to improve their lives. The process of gathering data using a middleware to publish it on the web is a common case of study. The motivation of using heterogeneous data capturing devices, the creation of complex systems that formalize and present the acquired data in an analytical way, and the possibility of sharing the data using the paradigm of Linked Open Data, are motivating the creation of different frameworks that enrich the extraction and management of data, to be subsequently shared over the web in a semantic notation.

In Calbimonte et al. (2014), authors present a middleware that aims to create a standardized framework to gather data from virtual sensors and annotate it using semantics and adding meaning/context, to be later provided and shared under certain restrictions. The authors describe the devised architecture and the usage of a preconfigured Ontology in creating a standardized semantic data sharing tool. In Le-Phuoc et al. (2012), authors presents another middleware solution to acquire data from IoT sensors, databases, RESTful applications and Semantic Web, and transform it into Linked Data Streams, to be made available and consumed by third parties through a Web interface and a SPARQL endpoint. Other different approaches use a data conversion process from relational data (i.e. a database) to semantic data using a preconfigured mapping file attached to a previously designed Ontology. For example in Langegger et al. (2008), authors present a middleware to provide an access point to the stored data from different sources, like relational databases or CSV files.<sup>2</sup> The system contains a SPARQL based endpoint which receives requests from the end users, and a middleware performs the needed conversions to translate the performed queries to non-semantic data. In our system, we have followed a similar mapping approach, but we have extended it, providing a semantic reasoning mechanism that not only maps the data to the target ontology but also extends it by adding new inferred knowledge.

Other approaches involve capturing data through IoT devices and managing the cognitive information through a middleware that stores the data directly in Resource Description Framework (RDF)<sup>3</sup> (Vlacheas et al. 2013). In Barnaghi et al. (2013), authors create a semantic framework to annotate streaming sensors data by using the semantic web to share geographic positions in real time. Bischof et al. (2014), describes a framework that allows to provide semantic meaning to the *smart city* data, connecting different applications. There are more approaches, such as the creation of a cloud of things by combining different IoT platforms, cloud computing and semantic data, to create different IoT middleware and visualization tools for gathered data (Petrolo et al. 2017); or a platform to design an Ontology based on a conversion of SPARQL Protocol and RDF Query Language (SPARQL) queries into Structured Query Language (SQL) queries, to gather data from relational databases with a semantic meaning, by using specific rules as a primary conversion tool (Nemirovski et al. 2013). We have decided not to follow a purely semantic approach, instead using a relational database and only mapping the acquired data to an ontology when it is accessed by third parties. This allows for a more efficient

and scalable system, as databases process large volumes of data faster than purely semantic frameworks.

There are several projects that have tackled the problem of developing middleware for *smart cities*. These projects aim to provide various services to help citizens to improve their well-being, and several of them are oriented specifically to senior citizens. For example, there are various projects focused on creating interventions based on the acquired data from IoT sensors, to inform about potential risk of decline or diseases. Generally, the data captured by the middleware are analysed to create personalized interventions and inform about health risk status, frailty onset and other issues. Some examples are: PreventIT<sup>4</sup>, FrailSafe<sup>5</sup>, Nestore<sup>6</sup>, AGNES<sup>7</sup>, STIMULATE<sup>8</sup>, eWall<sup>9</sup>, inCASA<sup>10</sup>, USEFIL<sup>11</sup> or CARE<sup>12</sup>. These projects do not provide a semantic endpoint that integrates all the gathered data, making it more difficult for third parties (companies, medical researchers, experts, etc.) to access it. The added value and the innovation in our system is that it has as an objective not only to help the citizens but also improve the research in frailty and MCI via an integrated and homogeneous system, that comprises and executes the conversion of stored data into semantic data using an elaborate and complex ontology, covering the various contexts of the citizens inside a city (places, actions, activities, behaviours, MCI and frailty data, indoor/outdoor spaces, etc.), and provides endpoints for SPARQL queries to access to the data. In addition, the system contains a fully integrated set of customizable interactive graphical interfaces (dashboards) and supporting RESTful middleware APIs that perform the needed data transformations and predictive analysis to provide data in detailed context to the geriatric/medical expert and caregivers.

There are other Smart City projects which use Linked Open Data to enrich the stored data with a semantic meaning, so it can be consumed easily by third parties. For example, Smart Santander<sup>13</sup>, SMART Odense<sup>14</sup> and SCRIBE<sup>15</sup>. The ACTIVAGE<sup>16</sup> project aims to achieve a wide-scale

<sup>2</sup> <http://www.ietf.org/rfc/rfc4180.txt>.

<sup>3</sup> <https://www.w3.org/RDF/>.

<sup>4</sup> <http://www.preventit.eu/>.

<sup>5</sup> <https://frailsafe-project.eu/>.

<sup>6</sup> <https://cordis.europa.eu/project/rcn/211703>.

<sup>7</sup> <http://www.aaleurope.eu/projects/agnes/>

<sup>8</sup> <http://www.aal-europe.eu/projects/stimulate/>

<sup>9</sup> [https://cordis.europa.eu/project/rcn/110560\\_es.html](https://cordis.europa.eu/project/rcn/110560_es.html).

<sup>10</sup> <http://www.incasaproject.eu>.

<sup>11</sup> <http://www.usefil.eu>

<sup>12</sup> <http://www.aaleurope.eu/projects/care/>.

<sup>13</sup> <http://www.smartsantander.eu/>.

<sup>14</sup> <https://www.odense.dk/byens-udvikling/smart-city>.

<sup>15</sup> [https://researcher.watson.ibm.com/researcher/view\\_group.php?id=2505](https://researcher.watson.ibm.com/researcher/view_group.php?id=2505).

<sup>16</sup> <http://www.activageproject.eu/activage-project>.

**Table 1** Differences between our approach and related projects

Project name	Indoor	Outdoor	City-wide	LD support	Reasoner support
PrevenIT	Yes	Yes	Yes	No	No
FrailSafe	Yes	Yes	No	No	No
Nestore	Yes	Yes	Yes	No	No
AGNES	Yes	No	No	No	No
STIMULATE	No	Yes	Yes	No	No
eWall	Yes	No	Yes	No	No
inCASA	Yes	No	No	No	No
USEFIL	Yes	No	No	No	No
CARE	Yes	No	Yes	No	No
S. Santander	Yes	Yes	Yes	Yes	No
S. Odense	No	Yes	Yes	Yes	No
ACTIVAGE	Yes	Yes	Yes	Yes	No
Our approach	Yes	Yes	Yes	Yes	Yes

semantic interoperability specifically among IoT platforms for Active and Healthy Aging in large-scale deployments. These projects use ontologies to map their stored data from different sources to augmented the semantic information, but they do not use a semantic reasoner to try to apply elicitation rules that allow to expand the stored knowledge. Our approach uses a rule engine based reasoner and applies different sets of rules that improve the stored knowledge and discover new statements that can refine the understanding of the acquired data.

Table 1 outlines the differences between different approaches adopted by other authors and our approach. The first two columns (*Indoor*, *Outdoor*), depicts if the system has support to collect data inside and outside a building (for example, we assume that a transport is an outdoor element to collect data). The following column (*City-Wide*) depicts if the system is capable or not to be deployed in a city-wide context level. The next column (*LD support*) illustrates if the system has or not support for Linked Data (convert the stored data into semantic data and share it over the web). The last column (*Reasoner support*) illustrates if the system has or not support to make a data reasoning process and try to predict new statements that can expand the stored semantic knowledge by using a set of rules.

The presented approaches have different features which makes them valuable in different scenarios. However, they do not cover all the features that our approach can perform. For example, PrevenIT is a system which uses the mobile technology (smartphone and smartband) to try to identify risk factors such as mild cognitive impairment and frailty. PrevenIT gathers data from users in a outdoor and indoor places to create an intervention system which can detect if a user has low health conditions and needs interventions to

improve its life. The solution is capable to be used in a city-wide context since the system could be scalable to cover more elderly citizens. However, the system is not prepared to use the Linked Open Data principles. The use of Linked Data provides the ability to share semantic data to third parties. This data can be used to create new policies which can adapt the citizens' needs and promote the inclusion of the elderly citizens.

FrailSafe is a project which combines medical and technological objectives to create a complete system which tries to cover various objectives: (1) understanding the frailty and develop measures to define it; (2) use the gathered data to present future outcomes; (3) develop useful tools to assist to elderly citizens; (4) create recommendations to prevent frailty. The project gathers individual data using different devices such as movement sensors, GPS tracker or smartphones. However, FrailSafe is focused only in monitoring each individual and create personalized interventions rather than create a complete city-wide context based system. In addition, this approach do not use Linked Data as an option to share data to external users and use it as a case of study to prevent frailty.

Nestore is a project to develop an innovative, multi-dimensional, personalized coaching system to support healthy ageing. The core idea of the project is to support elderly citizen by giving some hints and suggestions to have a healthy life based on his every day life activities. The project uses ICT solutions to extract data from outdoor and indoor places and it has a potential to be scalable to a city-wide context. In addition, it provides an interesting approach in which a virtual personal coach is created to determine what are the best lifestyle of a elderly person but do not use the Linked Open Data principles to share the obtained data from the citizens. Thus the data obtained is not shared to be used by researches or companies to develop new solutions or to use it as a case of study.

AGNES project uses a basic ICT platform to extract data from elderly citizens in a outdoor places. The core idea of the project is to create a web-based social network to stimulate the elderly citizens lives. The system gathers data from the users and shows this information in a easy way to be interpreted by informal carers, friends and family members. This data is used to know the current conditions of the monitored elderly users. The differences between this approach and our approach is that: (1) it gathers only indoor data, ignoring outdoor data (for example, when the citizen is shopping); (2) it is not prepared to be used in a city-wide context because it is not gathering data from outdoor places and it do not contain a large deployment structure: (3) it do not uses Linked Open Data nor a rule engine based reasoner to create semantic information.

STIMULATE project aims to create an innovative platform to provide electronic services to help elderly people

select the best travel itinerary based on their current capabilities. The platform uses a mobile device to offer assistance to elderly citizens and help them into their travels. This approach captures data in outdoor places but it is not prepared to capture it in indoor places. In addition, the system is not prepared to be used in a city-wide context nor Linked Data support. The project is interesting to obtain assistance in travelling but it is not prepared to improve the elderly citizen lives by using their regular day to day iterations.

eWALL is a project which creates prefabricated wall which show elderly person health status. The core idea of this project is to extract health conditions of a citizen in an indoor environment using ICT solutions and improve the elderly citizen independent living. The eWall project shows clinical information such as cardiopulmonary conditions, neuromuscular movements and so on. The project gathers data from elderly citizens in a indoor place and can be scalable in a city-wide context. However, the project is not considering the outdoor spaces as a other source of useful information. In addition, the project is not using Linked Open Data or using a generic rule reasoner to try to extract new semantic statements from acquired data.

inCASA project aims to use ICT technologies to monitor an elderly person in its home environment. This project monitors the daily live of elderly citizens which are living alone to profile a user behaviour. Using this profile, the system can achieve the creation of integrating health solutions and implement customized intelligent alerts services to improve the elderly citizen live. This approach is mainly focused in acquiring data from elderly citizen while they are at home, but the project is not ready to be exploited in a city-wide context and it is not prepared to have Linked Data support. In addition, the project is limited only in a home environment because it do not gather data when the user is outside.

USEFIL is a project similar to inCASA but using only open-source based technologies, avoiding the retrofitting of the users' residence and using low-cost ICT solutions. The project monitors the elderly daily live in a unobtrusive way to provide personalized services based in the user behaviours. This approach has the same lack of features that the inCASA project, it is not using any type of Linked Data solution and it is not monitoring the users when they are outside their home (outdoor monitoring).

CARE is an active and assisted living project which provides an indoor monitoring to create automatic alarms which inform of critical situations, for example a fall detection. This project creates an automated and intelligent environment to detect potential critical situations of an elderly citizen. The project only uses stationary technologies and do not use wereable devices to monitor the elderly citizen while it is outside, thus the project only gathers data in an indoor place. The project has the needed features to be scalable in

a city-wide context but only in indoor places with a specific monitoring devices. Compared with our approach, this project do not uses Linked Open Data to share the obtained results to third parties and create a collaborative knowledge.

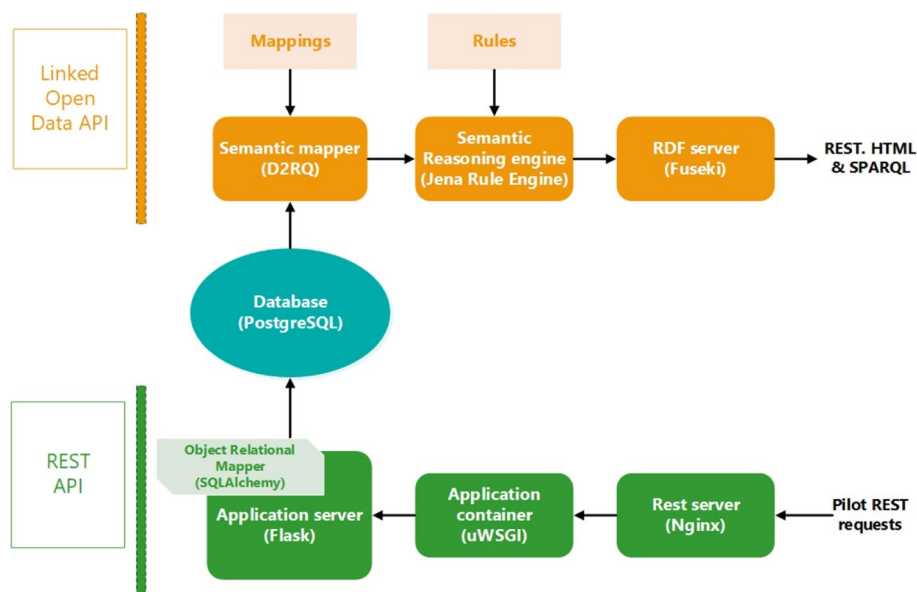
Smart Santander approach presents a complete solution of unobtrusive and obtrusive data collection in a city-wide context. This project uses huge range of different IoT solutions from movement sensors in outdoor spaces to NFC or QR codes in outdoor spaces (bus for example). In addition, this approach contains a module which gives to the stored data a semantic meaning and make it available to third parties. This approach demonstrate that is is possible to deploy a complete system in a city-wide context to monitor citizen daily lives, obtain useful information about their behaviours and share the acquired data as a semantic data. The difference between our approach and this project, is that Smart Santander do not use a semantic rule engine reasoner to infer new statements in the loaded semantic data. The use of a semantic rule engine reasoner is crucial in order to obtain new statements based on elicitation rules which creates new information statements to improve the obtained data by the sensors. Other difference is that our approach is focused on improving elderly ageing rather than Smart Santander which the main goal is to create a smart city in a generalist approach.

Smart Odense is an initiative of the Municipality of Odense and University of Southern Denmark to create an intelligent environment using the smart cities approach. The city contains a deployment of different sensors to monitor if a citizen uses the bike service or to have real-time information of free park lots or to improve the safety of the most vulnerable citizens. The main goal of this project is to improve the citizens life by creating new application services with the acquired data and provide better services to their citizens. This approach is similar that Smart Santander but it only gathers data from outdoor environments and it do not use a rule engine reasoning support.

ACTIVAGE is a project which uses IoT platforms to create an open framework for providing semantic interoperability of IoT platforms for active and healthy ageing. This project creates a set of different IoT layers to obtain data from citizens in different cities and share it to third parties to be consumed. This approach tries to promote the independent living, the mitigation of the frailty and the preservation of the elder quality life and autonomy in smart environments. The project creates a framework to identify and measure the requirements of the elderly citizens and opens the gate to third parties to obtain this data to develop different solutions. This project is similar as our approach, it contains a system to acquire data and share it but it do not use a semantic rule engine reasoner to infer the obtained data.

The system presented in this paper combines the strengths of the different analysed projects and contributions. Like

**Fig. 1** Overview of the complete architecture of the proposed system. The architecture is divided in two main modules: the REST API and the Linked Open Data API



several of mentioned projects, our system gathers citizen data using the IoT approach, but it does not impose restrictions to the used IoT middleware, allowing the cities to reuse or leverage their existing sensory infrastructure. The City4Age architecture integrates heterogeneous data sources into a single cloud repository, and annotates the data using semantics or semi-supervised data labeling with structured assessments from the domain experts. This allows third trusted parties like geriatricians, medical experts and researchers to use it to improve their own research in frailty and MCI risks. Instead of directly using a RDF storing solution like a triple store, the proposed system stores the data in a relational database, which offers more scalability and flexibility. It only maps the relational data to the target ontology when a third party queries the SPARQL endpoint. In addition, our approach develops a complete set of fully operative analytical dashboards to represent and visualize the stored data. The dashboards allow to explore, analyse, interactively annotate and collaboratively interpret the data, in order to detect the risks associated with early stage MCI or frailty.

### 3 System architecture

As mentioned in Sect. 1, to address the identified challenges, we have created an architecture<sup>17</sup> able to manage and annotate via semantics the heterogeneous data gathered from multiple cities. In this section, we describe the internal system architecture which is divided into two different modules. The first module is the REST API, which integrates a set of

different technologies used to acquire data from different sources, store it in the cloud, and aggregate/transform it. The REST API is the entry point for the IoT modules deployed in the cities. Section 3.1 contains an in-depth explanation of this module. The second module is the Linked Open Data API, which enriches the stored data with semantic meaning. This module also uses a semantic rule reasoner to improve the stored information by inferring new statements. Additionally, this module also allows third parties to explore and query the stored data using a SPARQL endpoint. The aim of this endpoint is to allow third parties such as governments, companies or researchers to take advantage of the aggregated City4Age dataset. This module is described in Sect. 3.2.

Figure 1 depicts the architecture of the system, with the modules that compose it. The cities gather data using their own IoT sensing infrastructure and middleware. The developed system is flexible enough to generalize the information received from different middlewares and to integrate it in a single cloud repository. The cities use the REST API to send their data to the repository. Then the REST API handles the data and stores it into a relational database. Once the data is stored into the database, the LOD API processes the data, mapping it to the City4Age ontology and adding semantic meaning to it. This information is processed further using the semantic reasoning engine to infer new information, according to the rules defined by the geriatricians involved in the project. Finally it is served to third parties using an RDF server that provides REST, HTML and SPARQL endpoints.

All data transactions are encoded in *JavaScript Object Notation* (JSON) and they need to follow some guidelines to be accepted by the REST API. These guidelines, derived mainly from the Web of Topics (WoX) conceptual model

<sup>17</sup> [https://github.com/aitoralmeida/c4a\\_data\\_repository](https://github.com/aitoralmeida/c4a_data_repository).

**Fig. 2** JSON example code

```

1 {
2   "action": "eu:c4a:usermotility:enter_bus",
3   "user": "eu:c4a:pilot:lecce:user:12345",
4   "location": "it:puglia:lecce:bus:39",
5   "payload": {
6     "position": "urn:ogc:def:crs:EPSG:6.6:4326",
7     "sensorId": "sensorId:beacon_123456"
8   },
9   "timestamp": "2017-01-22 07:08:41.22222",
10  "rating": 0.4,
11  "extra": {
12    "pilot": "lecce"
13  },
14  "secret": "jwt_token_k"
15 }

```

approach (Mainetti et al. 2015), enforce uniform communications and establish a standard for each type of data that it is processed by the system. Figure 2 depicts a sample code in JSON of the captured data. The sample contains data about the location of the user, the type of sensor which detects the action, the uncertainty of detection (in percentage), the exact position of performed action and the name/identifier of the action (enter the bus). Additionally, there are some extra fields to add a security layer to the data. The reason behind the use of a relational database rather than a triple-store database (Serral et al. 2013) is because relational databases are faster and more scalable.

### 3.1 REST application programming interface

The aim of the REST API is to provide a solution to manage data from different cities and store it in an secure environment. The implemented architecture defines a mechanism to manage data in using three abstraction levels. Hence, it makes possible to define what level of abstraction needs each city to manage its citizens data. The API allows the cities to decide what kind of information needs to be stored into database. There are three abstraction levels, each of them defining the precision of the data to be stored. The lowest level are the actions of the citizens, which are stored using the *add\_action* method. Actions describe an elementary action performed by a citizen (e.g. enter\_bus), usually captured by a single IoT sensory device. This endpoint represents the lowest abstraction level handling mostly raw or very simple data. The second level of abstraction are the activities performed by the citizens (which are stored using the *add\_activity* method). Activities are composed by a sequence of actions (e.g. visit\_family is composed by leaving the house, taking the public transport and arriving to the target house). The third level of abstraction are the geriatric factors (stored using the *add\_measure* method). Geriatric factors model variations of citizens behaviour that allow to identify risks related to frailty or MCI. These geriatric factors have been defined by behavioural experts and geriatricians. Each factor

has a set of subfactors to represent different aspects. For example, a factor could be called “motility” with a subfactor of “walking”.

Figure 2 depicts the data in JSON format used in an *add\_action* call in the REST API. The information contained in the JSON will be stored in a defined database schema described later in Sect. 4. The implementation of the proposed solution is based on the Python programming language using the *Flask micro-framework* as the application server and the *uWSGI application container*

#### 3.1.1 Application server (flask micro-framework)

The application server (APS) is the central module of the REST API. It provides the following services: (1) a set of different endpoints which the IoT middlewares can use to send the collected data with different aggregation levels; (2) a logical verification process to check if the given data is valid and can be stored in the system and (3) an internal process that handles the acquired data and stores it in a persistent environment. To create the APS we have used the Flask micro-framework,<sup>18</sup> which is a Python library used for the creation of Web Services. Flask relies on Werkzeug, an implementation of the Web Server Gateway Interface (WSGI)<sup>19</sup> protocol. It offers a set of libraries which help in the manipulation of Web services and Web Servers.

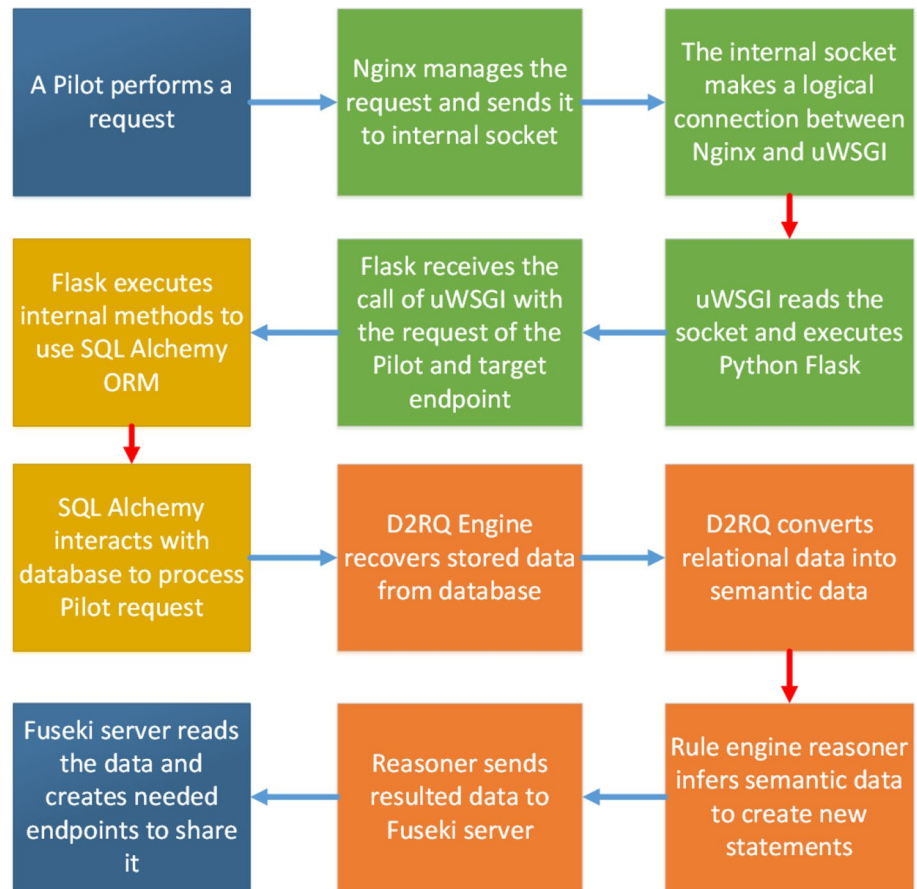
#### 3.1.2 Application container (uWSGI)

The application container (APC) is used to manage the executions of different programs. The main idea is to have a software which is capable of integrating a Web server and the APS. In this architecture, the APC acts as a middleware to handle the needed communications to pass the data

<sup>18</sup> <http://flask.pocoo.org/docs/0.12/>.

<sup>19</sup> <https://www.python.org/dev/peps/pep-0333/>.

**Fig. 3** Workflow of the proposed solution. The blue squares represent the input and output data. The green squares are part of the REST API workflow. The yellow squares are part of the database workflow. The orange squares are part of the Linked Open Data API workflow



from the IoT middlewares to the APS. To create the APC we have used uWSGI<sup>20</sup>, which is another implementation of the WSGI protocol. The difference between uWSGI and Flask micro-framework is that uWSGI works as a application container and creates an abstraction between the implemented code and a Web Server. uWSGI handles the data transmission between the Web Services and a reverse proxy based Web Server, for example, a Nginx<sup>21</sup> Web Server.

### 3.2 Linked open data interface

The LOD Interface is the module that gives semantic meaning to the stored data and shares it using several endpoints. The aim is to extract data from the relational database, map it with the City4Age ontology to create semantic data and then use a rule engine reasoner to infer new statements based on a certain rules to expand the knowledge. The rules need to represent the semantic context of the *Smart Cities* to fulfill the requirements of the citizens and create useful data. Figure 3 gives an overall view of the implemented solution. The

LOD Interface and the REST API share the same database. As can be seen in Fig. 1, data is modeled from database to the final output. This solution uses three sub-modules: (1) the mapping of the data is done using the *D2RQ* platform; (2) the rule engine inference program is done using the *Jena* libraries and (3) the endpoints to access to the final data are served by *Fuseki*.

#### 3.2.1 Semantic mapper (D2RQ platform)

The Semantic mapper (SM) is the sub-module in charge of giving semantic meaning to the relational data. The SM maps each table and each column of the relational data to the City4Age ontology, which describes the context of a Smart City. This mapping process is done by using an external file which contains a set of different description matches. The file indicates which are the Tables/Columns of the database that need to be directly converted to the Classes/Properties of the ontology. The SM processes each row of the tables and transforms it to RDF data. For example, if the relational data is extracted from the table “ExecutedAction”, the map file will contain a section which has the proper ontology description to give to each column of the table “ExecuteAction” the logical semantic relationships according to the

<sup>20</sup> <https://uwsgi-docs.readthedocs.io/en/latest/>.

<sup>21</sup> <https://nginx.org/en/docs/>.

**Table 2** An example of rules applied in the test

Name of the rule	“When” declaration
buildinglocation:	(?subject rdf:type vocab:location), (?subject vocab:location_indoor “t”), (?subject vocab:location_pilot_id ?object)
registeredstatus:	(?subject vocab:executed_action_date ?object) greaterThan(?object, “2014-04-21T07:08:41” 8sd:dateTime) lessThan(?object, “2014-08-21T07:08:41” 8sd:dateTime)
Direction of the rule	“Them” declaration
>	(?subject schema:Place city4age:Building)
>	(?subject schema:actionStatus “registered”)

Linked Open Data paradigm. Thus, for each row the mapper will create the RDF version of it. To create the SM, we have used the *D2RQ* platform.<sup>22</sup> The *D2RQ* platform is a set of tools written in Java, used for accessing relational databases as virtual read-only RDF graphs. This provides a solution to extract data from a relational database and share it by using a variation of the Pubby server<sup>23</sup> called D2R. In the implemented solution, D2RQ plays two roles. First, the extraction of relational data by using the D2RQ Mapping file to attach the data to a previously designed Ontology, which uses the D2RQ mapping language (Bizer and Seaborne 2004). Second, the use of its API based on the *Jena* libraries to extract and manage the loaded knowledge to use a rule engine reasoner and infer new statements.

### 3.2.2 Semantic reasoning engine (Jena)

The semantic reasoning engine (SME) sub-module processes the semantic knowledge to extract new information. When the SM builds the knowledge using its mapping file, the SME performs a reasoning process to try to discover new statements based on a set of elicitation rules. To create the SME, we have used the *Apache Jena* framework.<sup>24</sup> *Apache Jena* is a framework used to build applications based on Linked Data. In addition, these libraries support semantic inference based on four different types of rule engine reasoners. From Jena, we used its generic rule engine reasoner with a rule file based on a set of forward rules that help eliciting new knowledge (i.e. spatial and temporal knowledge, context related knowledge, etc.).

### 3.2.3 RDF server (Fuseki)

The RDF server is the final sub-module of the LOD Interface. This sub-module creates a set of different endpoints to allow

third parties to consume the inferred knowledge produced by the SM and the SME. The RDF server is capable of handling the requests of different users and provides an endpoint to accept SPARQL based queries. The core idea of this module is to provide the potential third parties a set of different web services to consume the previously extracted semantic data by the other parts of the LOD architecture. To develop the RDF server we have used Fuseki. It provides REST-style SPARQL HTTP Update, SPARQL Query, and SPARQL Update using the SPARQL protocol over HTTP. The reason to use Fuseki to serve Linked Data is because the implementation of D2RQ server (*D2R server*) is only designed to be a read only server. In our approach, we are loading data and generating new knowledge, thus it is necessary to use an external tool to share this new data due to the limitations of D2R server. The Fuseki server comes with a tool called *Shiro*, used to create some protection rules. This tool allows us to configure some security measures to have a control of the loaded data and the users who can access to the server.

## 3.3 Operation logic

Figure 3 depicts the workflow of the system. The City4Age project Ontology<sup>25</sup> is used to have the semantic representation of the context of the cities. Using this Ontology we created the file called *mapping file*. The mapping file is used to transform the data in the relational database to semantic data in OWL/RDF. In addition, we have created a set of rules to infer new statements using the rule engine reasoner, following an approach that we used previously in Almeida and López-de-Ipiña (2012b). These rules (see an example of the rules in Table 2) enrich and expand the existing knowledge, providing a more rich representation. The first rule in the example checks the locations of the data and detects if each location has an associated pilot and if it is registered in an indoor place. If these requirements are satisfied, then the rule is executed and it creates a new statement to represent

<sup>22</sup> <http://d2rq.org/getting-started>.

<sup>23</sup> <http://wifo5-03.informatik.uni-mannheim.de/pubby/>.

<sup>24</sup> <https://jena.apache.org/>.

<sup>25</sup> <https://morelab.deusto.es/ontologies/city4age>.

that the subject is inside a building; combining two different Ontologies (Schema<sup>26</sup> and City4Age). The second rule obtains the subject of an executed action (*add\_action*) and checks if the registered action is between two defined dates. If the rule is satisfied then the rule engine creates a new statement giving to the subject a new status (registered).

Once the data have been semantized and processed with the semantic rule engine it is ready to be queried by third parties. This is done by using an RDF server that provides SPARQL, HTML and REST interfaces to access the semantic data. One of the drawbacks of our approach is that the semantic data needs to be updated when the database changes. These changes should be reflected in the semantic knowledge loaded into the Fuseki server. The implemented solution to this problem is to repeat procedure periodically. Using this approach the knowledge is destroyed and loaded again with the updated data. The advantages of this solution is that we can modify the rules file to change the behaviour of the rule engine reasoner and adapt the knowledge to the requirements of the citizens without restarting the server.

### 3.4 System security

One of the main tasks of the system is to protect the privacy of the citizens and the security and integrity of their data. This has been done by implementing a set of security measures. First, the system has a set of security measures in the REST API to allow only authorized users to send or recover personal data. These measures are based in HTTP-Auth<sup>27</sup> identification and JSON Web Secured Tokens<sup>28</sup> to send encrypted credentials and allow or deny the users' requests. In addition, each authorized user in the system has a role-based access, thus it can only view or modify the data according to its permission level. Secondly, all the users' requests are protected using Secure Socket Layer (SSL) connections, with trusted signed certifications to ensure that every request to the API is encrypted and protected. The SSL connections create an encrypted tunnelled channel to protect the communication between two machines and make the information unreadable. Thirdly, the system uses two different encryption algorithms (bcrypt and AES), which makes the stored data unreadable to external attackers. Only the system contains the needed encryption and decryption algorithms to manipulate the stored data and make it readable. Fourthly, the Linked Open Data implementation contains a security tool called Apache Shiro<sup>29</sup> that provides a set of

security measures which restricts the access into the Fuseki server. This allows to prevent external SPARQL injections from malicious attackers (Orduña et al. 2010).

## 4 The City4Age data model

The City4Age data model has been designed to model the context of the *Smart Cities*. The cities can decide which information they want to store into the system by giving them a set of different levels of abstraction. These levels are designed to let cities decide what kind of information they want to send and store into the system. The designed database contains two different schemas, each of them is designed to store different types of citizen personal data: (1) the *Activity recognition* schema, which stores data gathered from different sensors and the inferred action and activities; (2) the *Shared repository* schema, which stores medical measures provided by expert systems and external users (geriatricians, medical researchers and so on). The system implements two schemas to isolate behavioural and medical data. The behavioural data is more related to psychology science whereas the medical data is more related to health science. These two logical schemas help in the detection and gathering process of the data by other external tools that need to interact with the database.

The *Activity Recognition schema* is designed to store actions and activities performed by the citizens. The model uses a set of entities that allow to store what actions are performed in a defined time and place and what are the activities related to these actions. The actions are represented by the *ExecutedAction* entity. It contains the action name, the confidence ratio, the location of the action, the city, additional information and, optionally, an activity attached to it. The *Activity* entity represents a collection of *ExecuteAction*-s. An example of the described entities could be "put the milk in a bowl" *ExecutedAction* that is part of the "Prepare breakfast" *Activity*. These two entries have their measures and they give different information according to the level of abstraction. In addition, there are more entities that store additional parameters derived from the previously described entities, whose purpose is to make data more scalable and organized. The design of this schema covers the necessity of storing the actions performed by the citizens in a defined spatio-temporal context.

The *Shared Repository schema* stores medical conditions of the citizens provided by the medical experts. This part of the model stores different types of physiological measures that allow to ascertain the risks related to frailty and MCI. The medical experts use these risks to recommend meaningful interventions. The model uses the *NumericalIndicator* entity to store the medical measures gathered by experts. The *DetectionVariable* entity stores the information of the

<sup>26</sup> <http://schema.org/>.

<sup>27</sup> <https://developer.mozilla.org/en-US/docs/Web/HTTP/Authentication>.

<sup>28</sup> <https://jwt.io/>.

<sup>29</sup> <https://shiro.apache.org/>.

condition that has been detected in a citizen and the *Care-Profile* entity stores data about physiological measures. The information given to this part of the model is generally based on the experts opinion, that use dashboard functionality to send data to the REST API with their calculated measures' levels, which in turn indicates if a user would need an intervention when its health is getting worse.

## 5 Analytics dashboards

“Dashboards” are the main developed analytics components of the City4Age system exposed to the end-users, providing the Individual Care Monitoring Dashboards (IMD) that offer the following functionalities:

- Interactive visualization of behavioural data, focusing on representation of the aggregated data of a single selected elderly individual (geriatric Care Recipient—CR).
- A collaborative environment for expert assessments/ annotations of the data.

The IMDs supports the input of various caregiver observations and indications for the City4Age digital interventions, as well as of incentives and feedback on the effects and results of the interventions. They are mainly targeted to geriatric caregivers, primarily health-care professionals (geriatricians, general practitioners, intervention staff, etc.), to help them detect, contextualize, interpret and annotate behavioural changes of the elderly people subject to their care.

In the context of combined data- and knowledge-driven analytics and detection approaches, the dashboards present the main developed interactive environment for geriatricians and caregivers to generally access, analyze and precisely annotate or label specific granular cases of interest in a variety of heterogeneous data collected to identify behaviour changes through varied Smart City IoT and Open Data infrastructure. In the hybrid of unsupervised learning techniques and knowledge-based models chosen as optimal for exploration of such data, there is exploration potential still in mechanisms for automatic or semi-automatic labelling of characterized behavioral patterns that denote “risk” warning and alerting. The long-term goal of the stated features of the IMD is therefore to detect and contextualize as early and precisely as possible risky behaviour changes that may lead to onset of MCI/frailty, and thus generate a significant volume of case-specific labelled referent patterns, to precisely capture the granular domain expert knowledge and integrate it with the underlying analytics, for iterative improvement of ML and other risk assessment and prediction techniques

used. Two main opposite directions can generally be distinguished in the context of City4Age detection:

- (a) in a top-down structure there are over 30 defined Geriatric Factors (e.g. “Motility”) in the conceived hierarchical model, that caregivers consider in characterizing a behavior for identifying insurgence of MCI/frailty
- (b) in a bottom-up structure there are detailed data from sensors (detecting, for example, granular movements), that can be interpreted into actions (e.g. one step), that can be interpreted as an activity (e.g. walking), further evaluated as part of a Geriatric Factor/Sub-factor (e.g. Motility) (Franco et al. 2017).

The City4Age adopted “hybrid” combined knowledge-driven and data-driven approach attempts as well to bridge the gap between the two directions—utilizing data mining to obtain most value from collected data, supported and refined by ontology-based recognition to associate contexts. Data-driven methods and techniques work “bottom-up” in the context, discovering frequency and similarity patterns in the data and synthesizing higher-level behaviors and Geriatric Factors from lower-level Actions and Activities. Knowledge-based methods function work mostly “top-down”, classifying the acquired data records (cases, patterns) in the hierarchical category structure through multi-criteria decision making processes. With most successful data-driven recognition approaches generally being based on supervised learning techniques, and relying on significant volumes of annotated or labeled cases/patterns (mostly manually input by the experts) to be used for training different kinds of classifiers, the comprehensive intuitive data visualization and interactive structured assessment/annotation features of the Dashboards are augmenting, accelerating and improving the data labeling, as well as general data interpretation, contextualization and integrity validation processes.

### 5.1 Interactive collaborative environment

Main visualization elements are rich composite diagrams—combined multi-line and stacked bar diagrams and radar (Fig. 4) diagrams, showing aggregated time-series data—as identified optimally understandable and intuitive by the Project (and external) geriatrician experts. Once the desired CR is selected through list/search in preceding screens, the dashboards present the data acquired on the person in selected or predefined time period in a general top-down flow, from high-level normalized aggregated model features (overall frailty status, geriatric domains, factor groups), supporting the drilling down to show specific granular data (Sub-factor values, variation measures, activities...).

Diagrams typically show the decomposed influence of the underlying constituting “child” variable values on each of



Fig. 4 Individual monitoring dashboards—radar (factor morphology) diagrams

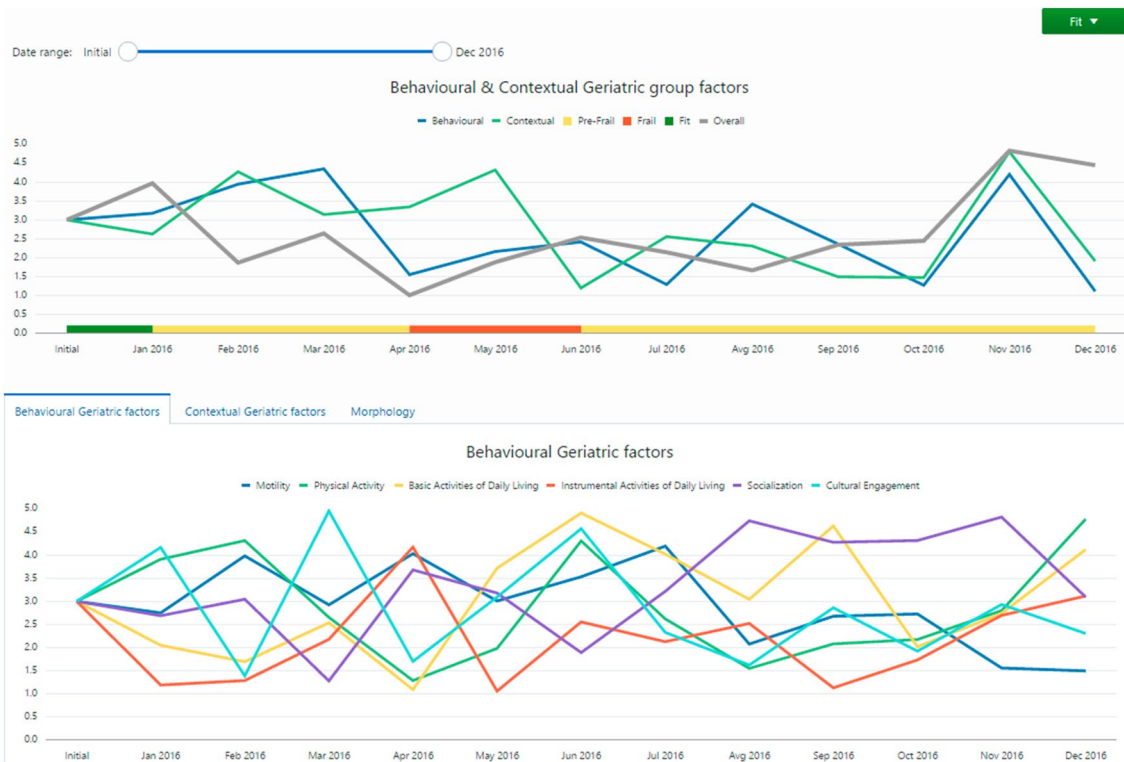
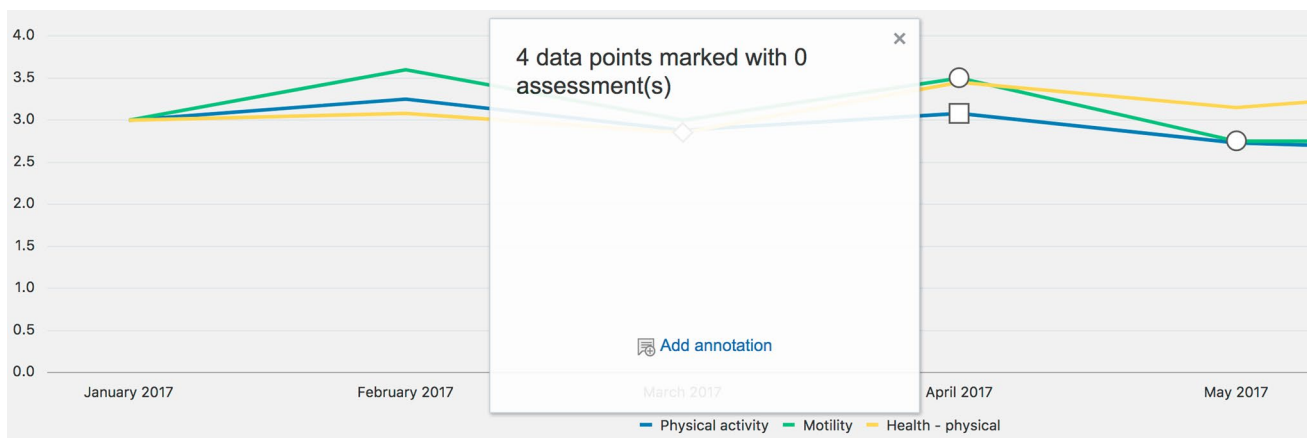


Fig. 5 Individual monitoring dashboards—combined multi-line and stacked bar diagrams

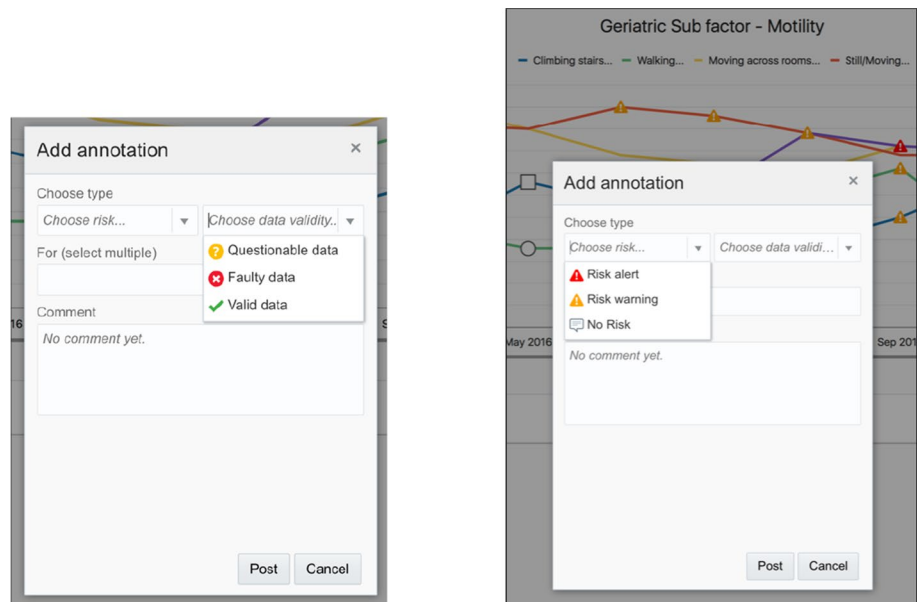
the detected variables (Geriatric Factor, Sub-factor, variation measure...) over time. The composite multi-line and bar diagrams can also show data on multiple model/aggregation levels at once, on a single diagram, as exemplified by the top diagram on Fig. 5—timeline changes of overall frailty status (in Fried Index notation Fried et al. 2001) are rendered as additional stacked bar below the multiple lines for each of the main geriatric domains (factor groups).

Basic interaction is a feature of the diagram UI elements (zooming in/out, showing/hiding specific series/groups, single/multiple point or window selection, etc.), and the innovative advanced interaction is provided by the custom component for interactive data assessments/annotations on graphs. An assessment can be assigned to each granular point on a diagram, or any set of points selected by multiple-click or window selection, via a modal pop-up panel for assessment



**Fig. 6** Adding assessment on a selected data point or dataset on diagram

**Fig. 7** Assessment/annotation input panel controls



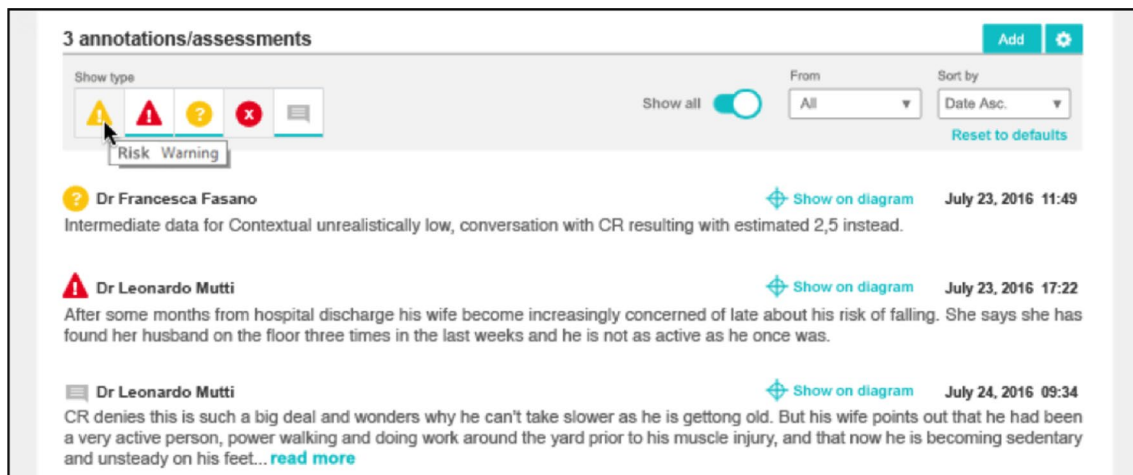
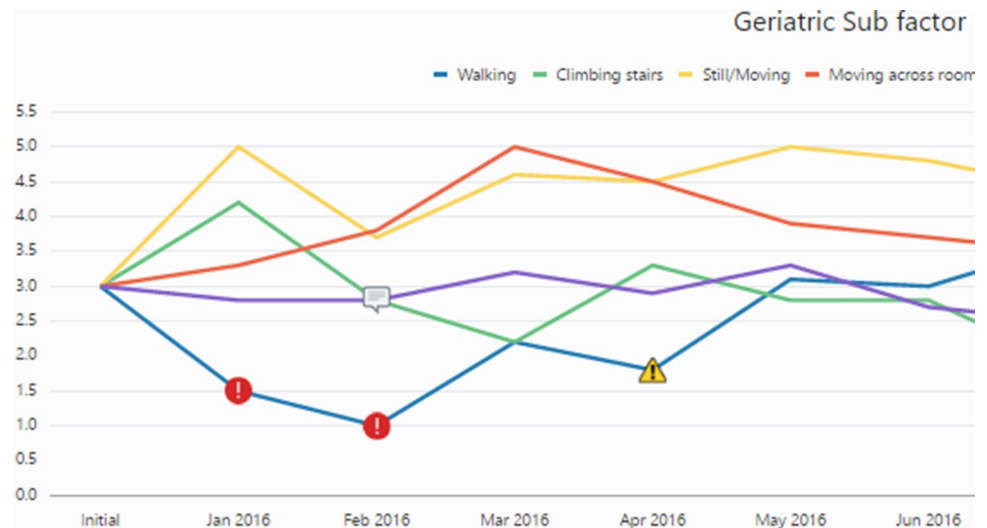
input, launched by the “Add annotation” command from the informative pop-up panel shown on cursor/pointer hover over selection (Fig. 6) on the diagram.

Each annotated data point or dataset can in turn, have one or a thread of multiple assessments assigned to it in different times. In common use cases of collaborative daily practice of caregivers, this supports the functionalities of:

- Accurately selecting
  - Specific individual peak outliers value point(s) on the diagram, likely to denote significantly deviating anomalous behaviour, or

- Sets of points (on one or different series on the graph) commonly marking relatively longer-term steady increase or decrease of corresponding variables or factors over weeks/months. This way, for example, on a diagram with decomposition of “Motility” Geriatric Factor, a geriatrician can select three average monthly values of “Still/Moving Time” in constant increase, together with one value of “Walking” Sub-factor in spike decrease in the same period. He/she can then assign a single assessment to these total four selected important values of two different variables, attributing them to one same phenomenon and potential motility risk, and thus denoting their connection (temporal, locational, causal, or other).

**Fig. 8** Markers on points with assigned assessments/annotations on diagram



**Fig. 9** Summary view of all assessments on a selection or diagram

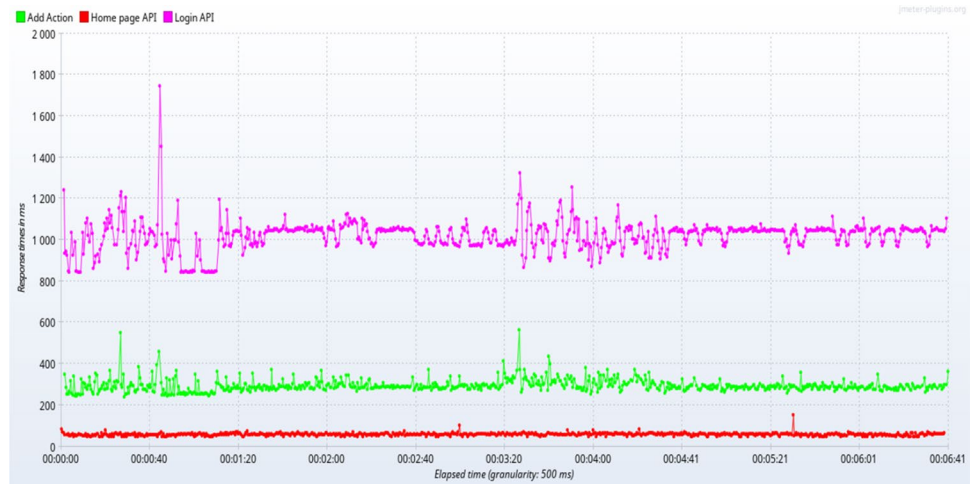
- Writing down and storing in the system annotations pinpointed to granular case-specific behavior change values and reading stored annotations (in free-form comments) provided by other colleague caregivers or different dashboard user roles. Common supported cases are writing indications for special attention to other caregivers, or indications for initiating or adapting an intervention. One or more target audience roles can be selected for each assessment in the input form panel (“For (select multiple)” field, Fig. 7). This way, an assessment with instructions or recommendations for an intervention can be targeted by the author, for example, to intervention staff and informal caregivers, who are to perform the intended intervention.

## 5.2 Structured data assessments as knowledge integration mechanism

The custom assessment/annotation component enables annotating the selected data point or dataset with structured categorized or quantifiable attributes, interpretable and usable for training and refining the targeted automatic risk/attention assessment and alerting performed by the Analytics (practically labelling data on various temporal and aggregation levels). Primary in initial implementation is the basic risk assessment categorization (warning—potential risk, alert—evident risk), complemented by the detection confidence label, also crucially important to account for inherent variable reliability rating of sensory acquisition data.

In case a caregiver finds the specific detected value(s) dubious or inconsistent, or evidently wrong (determined

**Fig. 10** Response time of API in the stress test



by examining/interviewing the actual CR), due to specific sensor problem or interference, the problematic data can be categorized as questionable or faulty via assigned assessment (Fig. 7). This data validity rating is used in the analytic framework, in the calculation of evaluated risk ratios, modes and priorities denoted and/or assigned to the marked data values, according to the failure time and effect models for health/frailty domain.

Overall dataset annotation (labeling) process performed this way via the IMDs is still essentially manual but leveraged and accelerated by the optimized UX and the work flow that multiple caregivers in all pilot cities perform simultaneously daily, to attain a significant number of input annotations. Assessments of data validity by the dashboard users are of additional importance in the deployment, testing and piloting operation of the system, to capture and indicate to the project development and integration teams all eventual problems or faults in the sensory acquisition and machine estimations. Similar or same behavior variation pattern can denote high risk for one person but low or no risk for another, and it is of high importance to resolve and filter out the “false positives”—numerous transient variations caused by external environmental factors (unobservable heterogeneity, model “frailty” Ayyagari 2008) or sensor imprecision, not by onsets of MCI or frailty targeted for recognition.

The categorized labeling attributes feature additional data marker icons for each of the categories, and these markers are rendered on the source diagrams, so the annotated points can be seen immediately on the graph. By default the risk categorization markers are plotted, being of highest interest, but the relevant property is parameterized in the custom assessment component, allowing the setting of optional other categorization instead. For points that have assigned multiple assessments with different risk categorizations, the marker denoting highest criticality in the set is shown (e.g.

red alert icon with exclamation mark higher criticality than similar yellow warning icon on Fig. 8).

In case a user selection on the diagram comprises a data point or points that have assessments assigned, all assessments on selected points (or datasets the points belong to) are shown in the summary list view below the diagram, with expandable truncated comments, and a filter panel for easier management of long threads of assessments, and a repeated “Add” command (Fig. 9).

Similar ongoing efforts in development of advanced tools for visualizing large-volume heterogeneous healthcare data in integrated unified timelines can be found in recently published works and articles such as Ghassemi et al. (2015) and Braga et al. (2016); featuring interactive composite or multidimensional diagrams of the same or higher complexity and sophistication level as presented (particularly in the example efforts from MIT). The assessment component as developed in IMDs, however, is unique featuring such precise granular case-specific labeling feedback from the UI dashboards/controls to the underlying analytics.

## 6 Validation

To validate the presented system we have performed three different tests. The first one is a *stress test* to evaluate if the API can support heavy loads of data. This test ensures that the API is capable of handling multiple client connections and manage large amounts of data. The second one is a *black box test*. This test is done to evaluate if the implemented solution works as intended by sending a sample data and testing if a different type of rules infers new statements.

**Fig. 11** SPARQL sample query used in the *black box* test

```

1 prefix schema: <http://schema.org/>
2 prefix deusto: <http://www.morelab.deusto.es/#>
3
4 SELECT ?sub ?obj WHERE {
5   ?sub schema:actionStatus ?obj
6 }

```

**Fig. 12** Part of SPARQL results in *black box* test

```

1 deusto:executed_action/11
2 deusto:executed_action/6
3 deusto:executed_action/5
4 deusto:executed_action/12
5 deusto:executed_action/7

```

Then, the test checks if the inferred statements and actual loaded knowledge is sent to the Fuseki server and is available to query with SPARQL. The last test is a *real deployment* of the system in six cities (Athens, Birmingham, Lecce, Madrid, Montpellier and Singapore). The test shows the results obtained in the first deployment phase of the project. This test shows how the end-users are capable of storing large quantities of data and shows the first stage results of the City4Age project.

## 6.1 Stress tests

To perform the stress test, we decide to use JMeter<sup>30</sup> to simulate the REST clients sending information to the API. To do the experimentation, we created a random list of different *add\_action* samples. The resulting file contains a list of 30 random samples. To make this test more realistic as possible, we simulate 800 citizens sending information to the system in an intervals of 0.5 s. Each of them sends three different requests.

The first one is a GET request<sup>31</sup> to the main page of the API, which returns a message with a status code 200 to the user if all works as intended. The second one is a POST request<sup>32</sup> where the user sends its login credentials to be authenticated in the API. We assume that all users send valid user login requests, thus the system will return a request message to inform on the successful login with a status code of 200. The third one is a POST request where the user sends the list of 30 JSON instances of *add\_action* to be stored in the database. If the data is stored and the API can confirm the committed action, then it will return a 200 status code with a message to inform that everything is correct.

The simulation environment is a dedicated server based on a Intel(R) Xeon(R) E5606 processor at 2.13 GHz clock speed, with 8 GB of RAM memory at 1333 Mhz clock speed (in dual channel mode) and a 500 GB ATA disk with

<sup>30</sup> <http://jmeter.apache.org/>

<sup>31</sup> <https://tools.ietf.org/html/rfc7231#section-4.3.1>.

<sup>32</sup> <https://tools.ietf.org/html/rfc7231#section-4.3.3>.

**Table 3** Results of the rule engine output

Number of elements	Execution number	Inferred count
757	1018	54

maximum data transfer speeds of 300 MB/s and 7200 nominal media rotation rate.

The total execution time took 6 mins and 41 s to complete. We repeated this test five times to ensure that results were consistent. In all iterations, the stress tests was finished successfully without any errors. The results of the simulation are depicted in Fig. 10.

As can be seen in the results, the implemented solution can handle multiple massive requests in a short period of time. This confirms that the API can handle requests from different citizens without major problems. In addition, we can confirm that the API is well optimized when users need to enter new data into database, as depicted in Fig. 10. The login request takes more time than the other requests to be completed. This behaviour is expected because the implemented solution creates an encrypted cookie for each successfully logged client. In the following iterations, the user will be logged into the system, thus it would not require to login again and it only would need to send his/her data. The *Home page API* data contains the minimum execution time performed by a simple request to the APIS's home page. The differences between home page requests and add action request are minimum in terms of response time, hence we can confirm that the implemented solution works as intended and it can handle massive amounts of data from citizens.

## 6.2 Black box test

The *black box* test is focused in validating the LOD. To accomplish this part simulate the action of adding new data and then analyse if the rule engine reasoner infers new statements based on a set of rules. To test if data is loaded and inferred in the Fuseki server we send a SELECT SPARQL

**Table 4** Results of the first phase of the deployment

Pilot	Sensor measurements	Geriatric measures	Users
Athens	58,098	175,118	46
Birmingham	126	87,183	29
Lecce	48,157	250	27
Madrid	29,777	2,189	30
Singapore	54,932	8,285	14
Montpellier	162,538	13,783	19

query to evaluate the results. If the initial and inferred data are loaded properly, we can confirm that the rule engine reasoner is generating new knowledge based on our set of rules and this knowledge is being shared over the Internet. To validate the test results, we analyse the logging information generated by the rule engine reasoner to ensure that the data is correct.

The rules used in this test are the previously presented ones in Table 2 and the *add\_action* samples of Fig. 2. The SPARQL query statement sent to the server is showed in Fig. 11.

The SPARQL query results showed in Fig. 12 reveal that the data is loaded in Fuseki and is being shared over the Internet because the sentence returns the “then” declaration coded into the rule number two. In addition, we analysed the results gathered by the logs of the Rule Engine Reasoner. These results are showed in Table 3. In addition to the results in the table, the program logs give additional information such as successfully connections to the Fuseki server, a list of the generated RDF graphs and information about error handling, thus it is possible to confirm that the rule engine reasoner is doing the inference job correctly and it is not giving errors.

The results given by SPARQL query and the information given by the program logs confirms that LOD is loading data from database, mapping it to knowledge, inferring and serving it into the Web.

### 6.3 Pilot sites deployment

To test the presented system in a real environment, it has been deployed in six different cities around the globe: Athens (Greece), Birmingham (UK), Lecce (Italy), Madrid (Spain), Singapore (Singapore) and Montpellier (France). Each deployment has several senior citizens using the system, plus their formal or informal caregivers and the geriatricians. The systems allows each of the pilot sites to decide their integration granularity, either providing low level sensor measurements (which will be used to infer the users’ activities and behaviour) or high level geriatric measures. This flexibility has allowed to the participating systems to

reuse the IoT and sensor infrastructure already in place in each of the cities.

Table 4 shows the results at the end of the first phase of the system deployment in the pilot sites. The table shows both the low level (the sensor measurements) and high level (geriatric measures) information uploaded by each pilot, and how many senior citizens have taken part in the deployment. As can be seen in the table, some of the cities that take part in the pilots prefer to send directly the low level information to the system in order to be processed, while other pilots provide directly the geriatric measures. In the first case, the system will be in charge of processing that sensor data in order to find meaningful behaviour information. In the second case the geriatric measures will be directly used in the system dashboards.

## 7 Conclusion

In this manuscript we present the citizen data management and analysis framework developed within the European H2020 City4Age project. The project aims to promote Ambient Assisted Cities to improve the lives of their elderly citizens by using modern IoT infrastructures, machine learning techniques and the Linked Open Data paradigm. The motivation of the project is the early detection of risks related to Mild Cognitive Impairments and frailty to perform preventive interventions and improve the quality of life of the citizens.

We have made a review of the literature, analysing the different approaches used to create city-scale assistive environments. In this context, we have presented our approach which (1) gathers data from citizens using the city’s own IoT infrastructure and a REST application service that integrates that data using different abstraction levels; (2) enriches the gathered data providing semantic meaning, by matching it with a previously designed ontology; (3) infers new knowledge based on the stored data using a rule engine based reasoner and spatio-temporal rules and (4) shares the results over the web providing SPARQL and RDF endpoints. In addition, we present the Individual Care Monitoring Dashboards (IMD) infrastructure that provides a visual analytical representation of the users data to the caregivers. The IMD represents the behaviour data of a citizen to be analysed and interpreted by the expert geriatric caregivers. The data can be annotated in a structured or free commented manner with additional specifications, to relate the acquired data from the API with ground truths and validate their quality. Furthermore, based on the represented data, the caregivers can assess the risk condition of a citizen to perform the required interventions to improve his/her health conditions.

Finally, we have performed two validation tests to assess if the proposed tool works as expected. We have made two

different tests: (1) a stress-based test to know if the API can work in a real environment like a big city; (2) a *black box* based test to know if the proposed architecture is working as described. Additionally, we have deployed the infrastructure in six different cities, gathering data from their citizens and providing added values to them.

As future work we plan to extend the analytical dashboard functionality by adding automated algorithms that will assist the caregivers in the evaluation of the elderly citizens' behaviour. Early versions of these algorithms in Almeida and Azkune (2017) and Almeida and Azkune (2018); are already being tested on the pilot sites and will be providing additional information in the IMD.

**Acknowledgements** This work has been supported by the European Commission under the City4Age project Grant agreement number: 689731.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

## References

- Alawadhi S, Aldama-Nalda A, Chourabi H, Gil-Garcia JR, Leung S, Melloulou S, Nam T, Pardo TA, Scholl HJ, Walker S (2012) Building understanding of smart city initiatives. In: International conference on electronic government. Springer, pp 40–53
- Almeida A, Azkune G (2018) Predicting human behaviour with recurrent neural networks. *Appl Sci* 8(2):305
- Almeida A, López-de-Ipiña D (2012a) A distributed reasoning engine ecosystem for semantic context-management in smart environments. *Sensors* 12(8):10208–10227
- Almeida A, López-de-Ipiña D (2012b) Assessing ambiguity of context data in intelligent environments: Towards a more reliable context managing system. *Sensors* 12(4):4934–4951
- Almeida A, Azkune G (2017) Activity recognition approaches for smart cities: the City4Age use case. In: Research and technologies for society and industry (RTSI), 2017 IEEE 3rd international forum on, IEEE, pp 1–5
- Almeida A, López-de-Ipiña D, Aguilera U, Larizgoitia I, Laiseca X, Orduña P, Barbier A (2009) An approach to dynamic knowledge extension and semantic reasoning in highly-mutable environments. In: 3rd symposium of ubiquitous computing and ambient intelligence 2008. Springer, pp 265–273
- Ayyagari P (2008) Preventive health behaviors among the elderly. PhD thesis, Duke University
- Azzara A, Bocchino S, Pagano P, Pellerano G, Petracca M (2013) Middleware solutions in WSN: The IoT oriented approach in the ICSI project. In: Software, telecommunications and computer networks (SoftCOM), 2013 21st international conference on, IEEE, pp 1–6
- Barnaghi P, Wang W, Dong L, Wang C (2013) A linked-data model for semantic sensor streams. In: Green computing and communications (GreenCom), 2013 IEEE and internet of things (iThings/CPSCom), IEEE international conference on and IEEE cyber, physical and social computing, IEEE, pp 468–475
- Bischof S, Karapantelakis A, Nechifor C, Sheth AP, Mileo A, Barnaghi P (2014) Semantic Modelling of Smart City Data. <http://corescholar.libraries.wright.edu/knoesis/572>
- Bizer C, Heath T, Berners-Lee T (2009) Linked data-the story so far. In: Semantic services, interoperability and web applications: emerging concepts, pp 205–227
- Bizer C, Seaborne A (2004) D2RQ-treating non-RDF databases as virtual RDF graphs. In: Proceedings of the 3rd international semantic web conference (ISWC2004), vol 2004. Springer
- Braga A, Portela F, Santos MF, Machado J, Abelha A, Silva Á, Rua F (2016) Pervasive patient timeline for intensive care units. In: New advances in information systems and technologies. Springer, pp 527–536
- Calbimonte JP, Sarni S, Eberle J, Aberer K (2014) XGSN: an open-source semantic sensing middleware for the web of things. In: TC/SSN@ISWC, pp 51–66
- Caragliu A, Del Bo C, Nijkamp P (2011) Smart cities in Europe. *J Urban Technol* 18(2):65–82
- Dey AK (2001) Understanding and using context. *Pers Ubiquitous Comput* 5(1):4–7
- Franco M, Sergio C, Giovanni R, Letizia V (2017) City4Age Frailty and MCI risk model, v 1. <http://www.city4ageproject.eu/index.php/download/d2-01-city4age-frailty-mci-risk-model-v1/>. Accessed 17 Dec 2016
- Fried LP, Tangen CM, Walston J, Newman AB, Hirsch C, Gottdiener J, Seeman T, Tracy R, Kop WJ, Burke G, McBurnie MA (2001) Frailty in older adults: evidence for a phenotype. *J Gerontol Ser A* 56(3):M146–M157
- Ghassemi M, Pimentel MA, Naumann T, Brennan T, Clifton DA, Szolovits P, Feng M (2015) A multivariate timeseries modeling approach to severity of illness assessment and forecasting in ICU with sparse, heterogeneous clinical data. In: AAAI, pp 446–453
- Hitzler P, Krötzsch M, Parsia B, Patel-Schneider PF, Rudolph S (2009) OWL 2 web ontology language primer. *W3C Recomm* 27(1):123
- Ji Z, Ganchev I, O'Droma M, Zhao L, Zhang X (2014) A cloud-based car parking middleware for IoT-based smart cities: design and implementation. *Sensors* 14(12):22372–22393
- Komninos N (2002) Intelligent cities: innovation, knowledge systems, and digital spaces. Spon. <https://books.google.com.uy/books?id=psQq2Pjp07gC>. Accessed 2 Mar 2018
- Komninos N (2009) Intelligent cities: towards interactive and global innovation environments. *Int J Innov Reg Dev* 1(4):337–355
- Krylovskiy A, Jahn M, Patti E (2015) Designing a smart city internet of things platform with microservice architecture. In: Future internet of things and cloud (FiCloud), 2015 3rd international conference on, IEEE, pp 25–30
- Langegger A, Wöß W, Blöchl M (2008) A semantic web middleware for virtual data integration on the web. In: European semantic web conference. Springer, pp 493–507
- Le-Phuoc D, Nguyen-Mau HQ, Parreira JX, Hauswirth M (2012) A middleware framework for scalable management of linked streams. *Web Semant Sci Serv Agents WWW* 16:42–51
- Mainetti L, Manco L, Patrono L, Sergi I, Vergallo R (2015) Web of Topics: an IoT-aware model-driven designing approach. In: Internet of things (WF-IoT), 2015 IEEE 2nd world forum on, IEEE, pp 46–51
- Nations U (2014) World urbanization prospects: the 2014 revision, highlights. Department of Economic and Social Affairs, Population Division, United Nations
- Nemirovski G, Nolle A, Sicilia Á, Ballarini I, Corado V (2013) Data integration driven ontology design, case study smart city. In:

- Proceedings of the 3rd international conference on web intelligence, mining and semantics, ACM, p 43
- Ojo A, Curry E, Zeleti FA (2015) A tale of open data innovations in five smart cities. In: System sciences (HICSS), 2015 48th Hawaii international conference on, IEEE, pp 2326–2335
- Orduña P, Almeida A, Aguilera U, Laiseca X, López-de-Ipiña D, Goiri AG (2010) Identifying security issues in the semantic web: Injection attacks in the semantic query languages. *Actas de las VI Jornadas Científico-Técnicas en Servicios Web y SOA* 51:4529–4542
- Petrolo R, Loscri V, Mitton N (2017) Towards a smart city based on cloud of things, a survey on the smart city vision and paradigms. *Trans Emerg Telecommun Technol* 28(1):e2931
- Serral E, Mordinyi R, Kovalenko O, Winkler D, Biffl S (2013) Evaluation of semantic data storages for integrating heterogenous disciplines in automation systems engineering. In: *Industrial electronics society, IECON 2013-39th annual conference of the IEEE*, IEEE, pp 6858–6865
- Shapiro JM (2006) Smart cities: quality of life, productivity, and the growth effects of human capital. *Rev Econ Stat* 88(2):324–335
- Theodoridis E, Mylonas G, Chatzigiannakis I (2013) Developing an iot smart city framework. In: *Information, intelligence, systems and applications (iisa)*, 2013 fourth international conference on, IEEE, pp 1–6
- Torres L, Pradas L, Pina V, Martínez V, Royo S, de Zaragoza Facultad de Ciencias Económicas y Empresariales U, de Comercio e Industria de Zaragoza CO (2005) E-government and the Transformation of Public Administrations in EU Countries: Beyond NPM Or Just a Second Wave of Reforms? *Documentos de trabajo*, Universidad de Zaragoza, Facultad de Ciencias Económicas y Empresariales. <https://books.google.com.uy/books?id=dKaQygAACAAJ>
- Vazquez J, Almeida A, Doamo I, Laiseca X, Orduña P (2009) Flexeo: an architecture for integrating Wireless Sensor Networks into the Internet of Things. *3rd Symposium of ubiquitous computing and ambient intelligence 2008*. Springer, Berlin/Heidelberg, pp 219–228
- Vlacheas P, Giaffreda R, Stavroulaki V, Kelaidonis D, Foteinos V, Poullos G, Demestichas P, Somov A, Biswas AR, Moessner K (2013) Enabling smart cities through a cognitive management framework for the internet of things. *IEEE Commun Mag* 51(6):102–111

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.