A Domain Model for Personalized Monitoring System Based on Context-Aware Data Fusion

Cristóvão Iglesias and Claudio Miceli
Programa de Pós-Graduação em Informática - PPGI / IM
Universidade Federal do Rio de Janeiro Rio de Janeiro, Brazil
Rio de Janeiro, Brazil
cristovao.casagrande@gmail.com and cmicelifarias@gmail.com

David Silva
Departamento de Ciências
Universidade do Porto
Porto, Portugal
daviddpssilva@hotmail.com

***Abstract*—Monitoring systems have been applied in many fields, but it is still challenging to design a monitoring system to deal with personalization of monitoring and unpredictability of events. We propose a domain model that is useful in designing real personalized monitoring system based on context-aware data fusion in the initial stage of the design phase of the development cycle. This domain model enables a clear understanding of the PMS domain and the specific vocabulary that comes with its facilities discussions among practitioners in the field. The domain model provides a conceptual perspective that shows all the important concepts which are needs to handle with personalization of monitoring and unpredictability of events in the domain of monitoring systems. Our approach is illustrated through domain modelling of a case.**

***Keywords*—Context, Data Fusion, Domain Model, Personalized Monitoring System**

I. Introduction

Due to technological advances and the emergence of the Internet of Things several monitoring systems can be developed. These systems have been applied in areas ranging from harbor monitoring systems [1] to health monitoring systems (HMS) [2]. A Monitoring System (MS) has the aim to allow users to capture data, to process and to disseminate information in a systematic way [3] for observing, checking, or keeping a continuous record of something. Generally speaking, one way to build a data-processing engine for a monitoring system is by using data fusion because of its three essential functions: data association, data alignment, and state estimation. Data Fusion in a more straightforward definition might be a process of combining data or information to develop improved estimates or predictions of entity states [4].

In recent years, the development of the context-aware fusion systems has been increased in order to improve the quality of the product and to offer tailored solutions to a specific field [4]. Successful context exploration can drastically increase the performance of a modern information fusion system: (i) Context improves the associativity between knowledge and models of the problem space, and observational data, enhancing the performance of the fusion, (ii) Context can be provided at different levels of the information fusion process and (iii) Context is an important source of semantics, which provides a means of linking data and models [2], [4]. Despite MS is being applied across multiple domains and enhanced with context-aware data fusion, the stakeholders when are developing monitoring systems based on context-aware data fusion, such as HMS, with traditional software development techniques, have to address two issues that concern to different phases of Systems Development Life Cycle (SDLC), including design, development, deployment, and maintenance, that are: (i) Personalization of monitoring [2], [5] and (ii) Unpredictability of Events [6].

Regarding Personalization of monitoring, the development of a Personalized Monitoring System (PMS) is still a problem that remains open in the literature [2], [5], because to monitor and to evaluate in a personalized way, for example humans, is a complex task. The following will be highlighted six points, related to humans, illustrating how this task is complex: i) Human behavior is subjective; ii) Complexity to perform several daily activities; iii) Behavior altering from one human to another; iv) Difficulty in defining a normal human behavior; v) Constant changes in human behavior and irregularity and vi) Anomalies vary depending on the nature and situation of the human. Although, it is difficult to perform a personalized monitoring, the PMS is required because a monitoring system focused on detecting general events (i.e. an event that occurs due to typical behavior from all individuals of a class) that may affect a Target Entity does not have accuracy to detect specific events (such as an event that can only occur due to the specific behavior of an individual in the class). Therefore, more efforts to improve the design and development of PMS for target entities, such as humans, in intelligent environments is needed in order to improve the accurateness and robustness of the results. They are still relatively distant from the real world, in which the behavior of a target entity may be mysterious and variable like a human.

Regarding Unpredictability of Events, we can say that the randomness in the real world generates many unpredictable or potentially harmful events that can affect a target entity, and therefore a monitoring system should change and adapt to better react [6]. Unfortunately, developing a system capable of detecting many events can cause expenses and it becomes very complex. An option would be to develop a Context-aware system [6] capable of handling as many events as possible with as few resources as possible. However, to the best of our knowledge, there is a lack of specifically tailored methods to address this problem in the design phase of the monitoring systems.

The right place in SDLC to address the two challenges mentioned above would be in the design phase because a PMS based on context-aware data fusion is a complex multi-faceted system, that evolves from many different disciplines. This different nature has lead to a mostly bottom-up approach to design of PMS where first are constructed the components, and the system-level issues addressed afterward. Thus the result is an ad hoc, prototype driven philosophy which is neither efficient nor effective [7]. We believe that the design of a PMS must be given proper consideration, with a tool of understanding and communication enabling the cooperation and interoperability between different stakeholders, potentially involved in very different application fields, often called silos [8], [9].

Generating a common understanding of the domain in question is the main purpose of a domain model, because it should provide a common lexicon and taxonomy, defining the main concepts and their relationships. A domain model needs to be stable; hence it needs to abstract from implementation details: it should separate out what does not vary from what does [9]. A useful domain model captures the essential abstractions and information required to understand the domain in the context of the current requirements, and aids people in understanding the domain (its concepts, terminology, and relationships) [8].

We offer an useful domain model in designing real personalized monitoring system based on context-aware data fusion in the initial stage of the design phase of the SDLC. This domain model providing a conceptual perspective that reveals all the important concepts as well as the associations among them that enables to deal with personalization of monitoring and unpredictability of events in the domain of monitoring systems. Our proposal is evaluated through the domain modelling of the a case illustrating the PMS domain model concepts.

The remainder of the paper is structured as follows. After describing the existing approaches in this field (Section II), we describe in detail our proposed PMS domain model (Section III). In Section IV, we present a way to apply the PMS Domain Model. This model is validated in the Section V, showing how the concepts in the domain model can be applied to concrete application. Finally, we conclude and give some outlook on future works (Section VI).

II. Related works

Some approaches of the literature could be used to design a PMS, but they have the following drawbacks. The [10] discusses an approach called JDL model in which it aims to develop a data fusion system. It comprises of five levels of data processing and one database, which are all interconnected by a bus. However, the JDL model is strictly a discussion aid and not an architecture or processing diagram and never really addressed the notion of exploration of contextual information in process design [11]. Nonetheless, the JDL model does not help in developing an architecture for a real system [10]. In [12] was proposed a general Data Fusion Architecture (DFA) based on the Unified modelling Language (UML) and using a taxonomy based on the definitions of raw data and variables or tasks and in [7] was proposed a Reference Model for Data Fusion Systems. With these two approaches, it is possible to design a monitoring system, but they are not expressive enough to allow the PMS design. These two approaches do not present context concepts that enable them to use it to detect an event. In [9] a domain model for IoT applications design was presented. This domain model can be used to design a PMS and some concepts of it are used in this present work. However, the points against the use of this proposal would be that it does not have context and Hard/Soft data concepts that are necessary in the data fusion process.

III. Domain Modelling Approach for Personalized Monitoring System

This section introduces the proposed domain modelling approach for personalized monitoring system based on context-aware data fusion. Before going further into details of the PMS domain model we will give our definition to PMS. Despite the considerable interest of the industry and the academia, there is no consensus about the definition of Personalized Monitoring System (PMS). Based on the definition of “personalization” as introduced by [13] and of "Monitoring Information System" as introduced by [3] we adopt the definition of PMS as

- *a system that allows users to capture data, process and disseminate information in a systematic way to perform actions such as detections, predictions or estimates about some event that may affect a target entity. In particular, these actions are performed from knowledge about the preferences and behaviors that a target entity can express as specific patterns that characterize it in an environment.*

From this definition was created the PMS domain model that is a UML class diagram [8] providing a conceptual perspective that presents all the important concepts in the domain of the PMS as well as the associations among these ones.

Next, we will detail the domain model that is based on eight core concepts:

- **User:** A PMS can be regarded as a technical system with some purpose, it is crucial someone to look at who handle it and how this relates to the system. Depending on the usage, users vary from human beings with dedicated functions to machines, devices, and services;
- **Environment:** A physical or virtual space that enables contexts to happen and it has the participation or it is performed by the Target Entity that inhabits or interacts with the environment;
- **Context:** A relevant situation that can be used to provide information either to condition expectations or to improve the understanding of a given inference or management problem [4];
- **Target Entity:** Any entity (e.g. car, human, animal, etc.) that can interact and express patterns in a context that can occur in a environment;

- **Event:** Any situation that makes a target entity to express anomalies in an environment context;
- **Device:** Technical physical component (hardware) with communication capabilities to other IT systems that collect hard data of a relevant situation.
- **Resource:** Computational element that gives access to data and information;
- **Service:** Services expose the Resources through a common interface and make them (as well as related Devices) available for applications and other Services.

In the following sections, these core concepts and their relationships are explained in details, including additional subconcepts.

A. User

By a PMS, Users can either be humans, e.g., a warehouse worker moving some pallet, or in Machine-to-Machine (M2M) scenarios, an Active Digital Entity as a program embedded in a manufacturing robot that is interested in personalized monitoring of a particular target entity. A PMS disseminates information for Users from Services. A User interacts with the PMS invoking or subscribing its services, such as alerts, notifications, summaries, etc. The User can be internal or external to the PMS. An internal user is the Target Entity that can request the PMS services as in the case of a patient in a personalized health monitoring system. Figure 1 shows the representation and relations of a User in the PMS domain model.

B. Environment

An environment can be physical or virtual. It is an essential element to design a PMS because by this it is possible to identify and to select the contexts that a Target Entity will participate and will interact. An environment is a space with appropriate conditions for a context to occur. For instance, a house (physical environment) contains others physical environments (living room, kitchen, bathroom, and bedroom), and allows contexts for a Target Entity such as consumption of water, electricity, and gas. A social network, which is an example of a virtual environment, enables contexts for a Target Entity such as text production, direct communication, etc. Figure 1 shows the representation and relations of a Environment in the PMS domain model.

C. Context

Context can be a data source categorized with dynamic or static [14]. Static Data Source may be available in static repositories such as maps, GIS databases, representations of roads, channels, bridges, etc. It can be represented as a Network Resource and used by a Data Fusion Resource. Dynamic Data Source may be available as dynamic data, such as meteorological conditions changing in space and time. Here, we will address context as a dynamic data source that can be available directly from both hard data sources (humidity or wind sensors, for instance) or soft data sources (such as human messages about environmental conditions) because the process of data fusion must take into account two types of data: soft and hard; in order to provide improved situational awareness and decision-making. As the context is a Dynamic Data Source, it provides information that can be expressed as dynamic variables [14]. We may distinguish dynamic variables in two types of variables used in inference: explicit problem variables and ancillary variables. Problem variables is a topic of interest defined as a set of state variables that an agent wishes to evaluate. A ancillary variable is a variable that the information exploitation system selects to predict, evaluate or refine an estimate of one or more problem variables [4]. By definition, one problem variable can serve as a ancillary variable for resolving another problem variable [4]. The relationship between Context and Event, shown in Figure 1, defines when the context will be used as a problem variable and ancillary variable. When a context is used as a situation of interest to provide constraints and expectations for constituent entities, relationships, and activities, it is a problem variable and the situation is used as a "context of". An example would be, "In the context of the present economic situation, we expect an increase in property crime" [4], [14]. When a context is used as a situation that is selected to help solve a problem it is an ancillary variable and the situation is used as a "context for". An example would be, "the town's economic situation provides a context for understanding this crime" [4], [14]. These two variables (problem and ancillary) and their relationships are used as data sources and rules by a Data Fusion Resource. The Problem Variable Rule and Ancillary Variable Rule are association classes derived from the relationship between Context and Event classes and the relationship that the context class does with itself, respectively. The Data Fusion Resource uses Problem Variable Rule and Ancillary Variable Rule in data fusion processes. As context can be expressed as a variable it can contain patterns and anomalies that can be used by the Data Fusion Resource to detect, identify, estimate and predict an event in a PMS. Figure 1 shows the representation and relations of a Context in the PMS domain model.

D. Target Entity

The Target Entity is shown in Figure 1. It is the central element of PMS because from this class that PMS will perform their services making estimates, predictions, detection through a *Data Fusion Resource*. A *Target Entity* interacts with a *Environment* participating or performing the Contexts that an environment allows them to occur. When a Target Entity participates of a context, it can express specific patterns because it may have a specific way of participating or interacting with a relevant situation. The expression of specific patterns by a target entity in an environment is the crucial point to do the personalization of monitoring. Because these specific patterns can characterize the Target Entity in the environment and they are used to detect anomalies caused by an event that may affect a target entity [15]. Thus, if a "Target Entity A" expresses a set of specific patterns when participating of a "Context X", the personalized monitoring for "Target Entity A" will occur according to these specific patterns that it expresses and will not be accurate for "Target Entity B", because it expresses another set of specific patterns when participating or performing the "Context X". It should be noted that specific patterns and anomalies occur in a dynamic context variable as detailed in the Context subsection. To illustrate, let us face the case of the elderly living alone. The elderly, like any human being, have specific habits to perform daily activities. The activities performed by them have a specific time and duration. Thus, activities such as waking up, having lunch, having dinner, have a schedule and a specific length of time for each elderly person, and this reflects on a specific pattern of domestic consumption for each elderly person. Therefore, domestic consumption is a context (a relevant situation that can be a dynamic context variable) and the domestic consumption pattern of a normal elderly person is different from an elderly person suffering with depression once this individual may present lack of appetite or insomnia that leads, for example, to an increase in the consumption of electric energy at dawn. Thus, different habits to perform daily activities generate different domestic consumption patterns. So starting from the principle that when a Target Entity inhabits or interacts with an environment, it expresses specific patterns in the contexts that participate. We have that the personalized monitoring of Target Entity can be done by detecting anomalies caused by some Events [15]. Still, in the example cited above, we have that personalized monitoring occurs because the use of domestic consumption as a context is a more realistic and more specific form of data collection for an elderly person. Figure 1 shows the representation and relations of a Target Entity in the PMS domain model.

E. Event

Event affects a Target Entity causing it to express anomalies when participating in a Context; in other words, an event makes a Target Entity to generate anomaly values in a dynamic context variable. If an Event does not happen, an anomaly does not occur as well and the dynamic context variable will continue to present values into a particular pattern. The concept of Event is essential in order to select the maximum number of situations that will be treated by the PMS using the same resource. Here the designer has to ask himself: "What is the maximum number of Events that the system being designed will be able to handle?" This answer comes from the contexts that the target entity will participate. This is a way of try to reduce unpredictability that the system will have to deal and optimizes the use of resources. Figure 1 shows the representation and relations of a Event in the PMS domain model.

F. Device

The Device is a piece of computing hardware and is the super-class for the more specific hardware. They are the components responsible for capturing the data generated by contexts in physical environments and that will be used by a Data Fusion Resource that will access them by invoking an On-Device Resource. Devices can be physically attached to Physical Entity (PE) (e .g., a tag), but they may also be in the environment of PE (e.g., presence sensor). Sensors allow the monitoring of PE's, whereas actuators can act on PE's. Tags uniquely identify PE's and can be read by sensors. Devices can be composed of Devices, e.g., a node in a sensor and actuator network may be composed of multiple sensors, actuators and general computing hardware. As a Device is part of the physical world, it is of course itself a PE. The relationships between Devices and PE's are shown in Figure 1.

G. Resource

Resources is a concept proposed in [9]; they are software components that implement certain functionalities, i.e., they provide information about Target Entity or allow the execution of actuation tasks and can invoke other resources. Their relations are shown in Figure 1. It can be of four types: On-Device Resource, Network Resource, Soft Data Capture Resource, and Data Fusion Resource.

- On-Device Resource (ODR) is resource hosted inside a device and enabling access to the device, e.g., Device driver, Programming API, data cache on gateway and data on an RFID tag.
- Network Resource (NR) are resource hosted somewhere in the network, e.g., cloud resources, Data repositories, EPCIS repository, and ERP database.
- Soft Data Capture Resource (SDCR) is a resource that collects soft information that is invoked by Data Fusion Resouce, e.g., Natural Language Processing (NPL) Application.
- Data Fusion Resouce (DFR) represents the PMS component that process multiple sources to obtain improved information about Event that can affect a Target Entity, e.g., JDL or Omnibus based applications. The DFR defines the Problem and Ancillary Variable that will be used in the fusion process through of use the association classes called Problem Variable Rule and Ancillary Variable Rule. The data flowing through these variables are collected based on their categorization as Hard and Soft Data. Thus, hard data are captured by Devices and soft data collected by SDCR. DFR accesses this data by invoking another resource as On-device Resource and SDCR. The DFR, besides invoking ODR and SDCR, invokes Resource Network because static context sources can be used from the Network Resource to help in the fusion process.

H. Service

The concept of the Service was introduced in [9] it exposes a standardized interface that can be invoked by the user. Resources provide the functionality, but may run on restricted Devices that may not be able to expose a suitable interface enabling users to interact with it. The Service has the internal knowledge how to access the functionality provided by a Resource and can do additional things like caching results, supporting additional interaction styles, e.g. push/pull, and help achieving scalability, e.g., through load balancing. Services can also be hierarchically structured, so that one Service may invoke other Services and combine the results. All relations regarding services are shown in Figure 1.

I. Limitations: Systems made from the PMS domain model

One limitation that stakeholders have to keep in mind is that the PMS made from PMS domain model can only do the detections or estimations from a period of data collection. The DFR only can make a detection or estimations if it has a pattern that is representative enough and therefore is necessary a great sampling in which it is made over a time period. From Target Entity's participation in contexts over a period of time, it is possible to obtain patterns from Target Entity that are meaningful and can be used to detect what will be an anomaly [15].

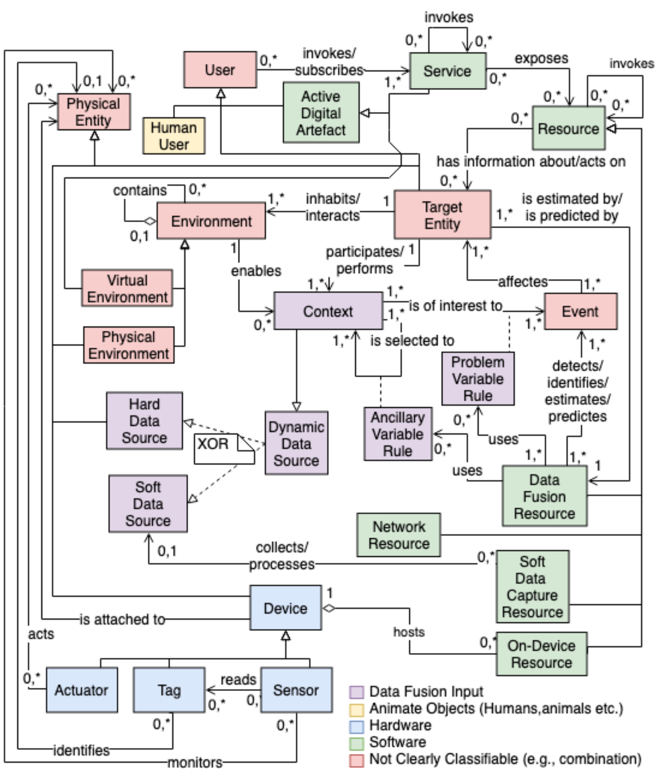

Fig. 1. PMS Domain model

IV. Applying the PMS Domain Model

As described above, the PMS Domain Model enables an evident understanding of the PMS domain and the specific vocabulary that comes with its facilities discussions among practitioners of the area. Furthermore, it also serves as a blueprint for analysts and architects when designing concrete, personalized monitoring systems. The typical design process will identify from the requirements the main elements of system. In terms of the Domain Model, this means that the following scheme will be followed:

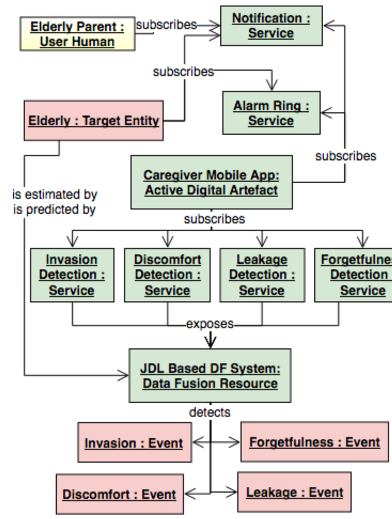

Fig. 2. Class instance diagram of case: Caregiver Mobile Application

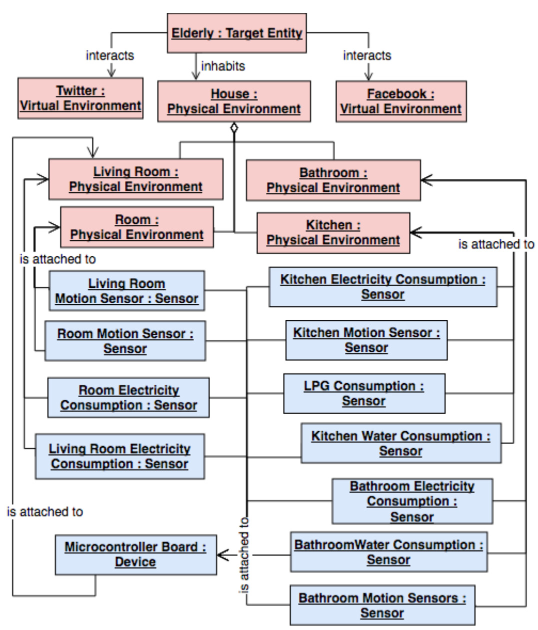

Fig. 3. Class instance diagram of case: Virtual and Physical Environments

1) Identify the Users of the system;
2) Identify the Target Entity;
3) Identify the Environment that Target Entity will inhabit or interact;
4) Identify the Contexts that Target Entity can participate in while inhabiting the Environment;
5) Select the Contexts that can be used to detect Events that may affect Target Entity;
6) Identify the Resources;
7) Map the Resources to the Device types.

Once this mapping is complete, it can be used as a basis for communicating within the design team and with the stakeholders. Analysts can then delve deeper into the details of the designed system by analyzing the all of the system's

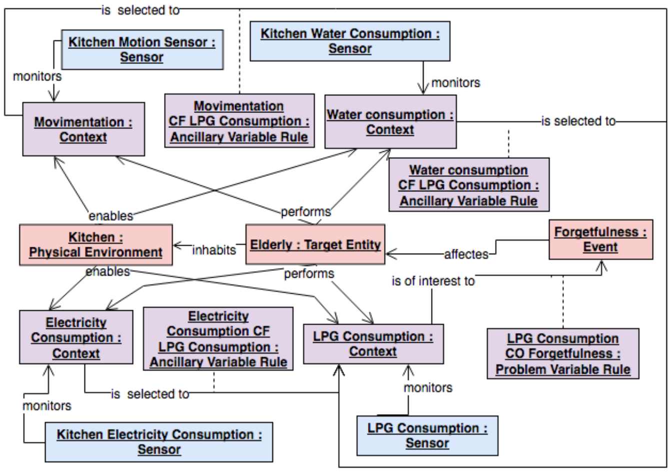

Fig. 4. Class instance diagram of case: Liquefied petroleum gas (LPG) Consumption Forgetfulness Detection

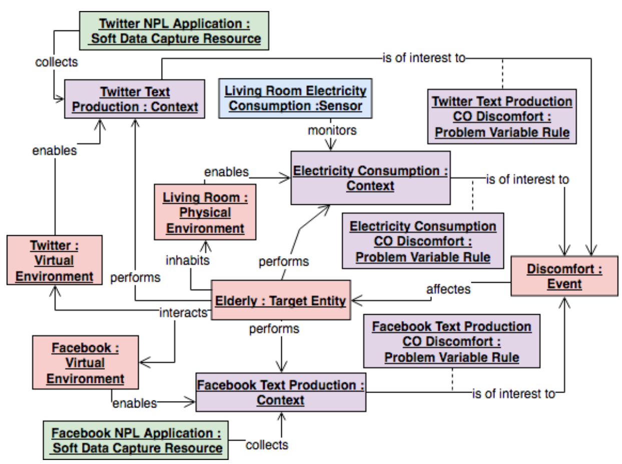

Fig. 5. Class instance diagram of case: Discomfort Detection

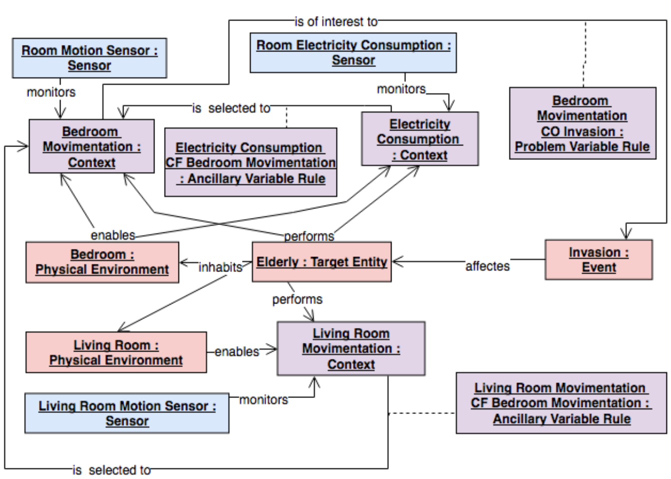

Fig. 6. Class instance diagram of case: Invasion Detection

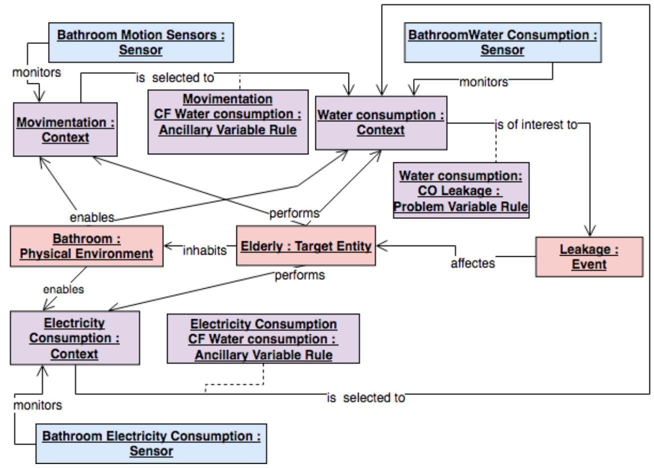

Fig. 7. Class instance diagram of case: Water Leakage Detection

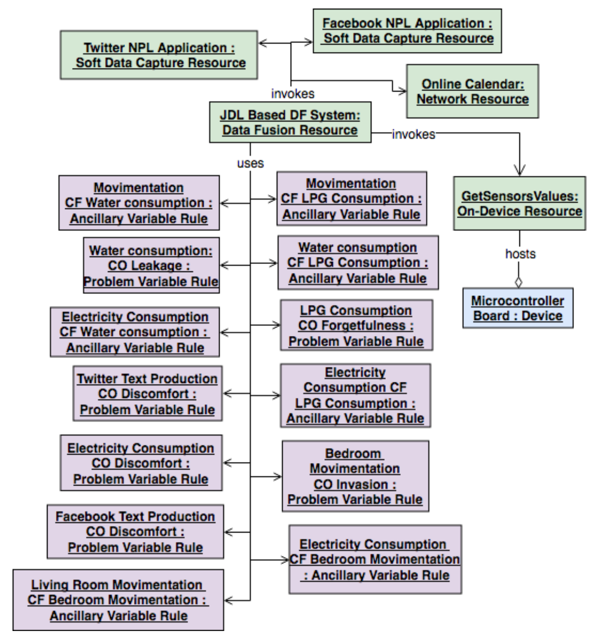

Fig. 8. Class instance diagram of case: JDL based Data Fusion sub-system

facets. This activity should be performed in parallel with the development of the system's communication, information and access control models of literature because they are tightly intertwined. An implementation of the System Functionalities (e.g. Look-up, Discovery, Resolution) is needed after PMS Domain models have been used to define the personalized monitoring system specific mappings. This implementation will cover the steps that are needed before the Endpoint-to-Endpoint communication, context exploitation for the data fusion process. Various architecture options are available for implementation purposes such as [4], [14], [16].

V. Case: Elderly Personalized Monitoring System

This section presents the case to validate the PMS Domain Model. Thereby, we identify the involved entities and the required mappings according to the process outlined in the previous section. The case is a Elderly Personalized Monitoring System, but PMS domain model can be applied to any problems domain. The instantiated domain model of the case is formed by seven diagrams illustrated in the Figures 2, 3, 4, 5, 6, 7 and 8 that will be detailed throughout the text.

A. *Defined Storyline*

Joanna is an elderly woman who lives alone and has a son called John. Since John cannot be with his mother on a daily basis due to work, he decided to use a system called Caregiver, a personalized monitoring of elderly people living alone to detect the following events: Invasion (because his mother lives in a dangerous neighborhood); Forgetfulness (because of her memory failures); Discomfort (because his mother stays long time alone) and leakage (to avoid unnecessary expenses). John and his mother use the system and receive notifications and alarms when the events described above are detected.

1) Forgetfulness detection of the LPG consumption from a User's Perspective: Joanna received her son's message at 10 pm. He said that he would go to her house for lunch at noon. Joanna would be out in the morning because she would have a medical appointment at 7 am, she decided to cook the meal at 10:05 pm to let it done for tomorrow. While cooking, her favorite TV show started. She went out the kitchen with the pans on fire and began to watch the TV show that will finished at 11:20 pm. Some minutes had already passed and she had felt sleepy. She turned off the lights and went to bed. She had forgotten that the liquefied petroleum gas (LPG) was been wasting itself. The system detected the unusual situation because it is not habitual to consume LPG without consuming electricity, water and without movement. Besides, by the Joanna´s consumption pattern, she had never consumed LPG for more than one hour. Consequently, the system rang the mobile alarm presenting the situation and reminding Joanna that she was still cooking. The modelling of this detection can be seen in the diagram of Figure 4.

2) Discomfort detection from a User's Perspective: John receives a weekly report about his mother's condition indicating discomfort with signs of depression. The system has detected this because his mother shown insomnia and negative feelings that are signs of depression. Joanna had an electricity consumption pattern that it did not exceed at 10:30 pm, but during the week she had spent a long time watching television until 4 am, wasting electricity, and much movement when trying to go to sleep indicating signs of insomnia. In addition to this abnormal electricity consumption, the system also detected that during this period, Joanna wrote many negative feelings beyond her regular posting pattern. The modelling of this detection can be seen in the diagram of Figure 5.

3) Invasion detection from a User's Perspective: Joanna left her house to go to the doctor at 7 am. However, 9 am was notified by mobile phone of an invasion at his house. The system detected this incident because it is impossible to go towards the Joanna´s bedroom without passing through the living room firstly. Electricity consumption usually occurs when someone arrives at home as well because the house front door is attached to the living room. Also, the system collected information from Joanna's online calendar that she would be away from home from 7 am to 11 am. The modelling of this detection can be observed in the Figure 6.

4) Water leakage detection from a User's Perspective: At 10 pm Joanna went to sleep, but at 1 am the alarm rang on her cell phone over water leakage in the bathroom. The system detected this because 1 am began to occur a consumption of water, without consumption of electricity and movement in the bathroom. Joanna usually consumes water and at the same time consumes electricity and has movement. The modelling of this detection can be seen in the diagram of Figure 7.

B. *Relation to the Domain Model*

The system encompasses (1) a set of devices endowed with sensors for capture movement and water, LPG and electricity consumption and (2) a mobile app called Caregiver which allows people receive alerts in anywhere (over the internet) thereby supporting mobility. People (son, relatives, neighbor, etc.) can sign up for receiving alerts whenever an event is detected.

1) Identify the Users and Target Entity of the system: The projection of the Users and Target Entity interaction with the system on the domain model is depicted in the class instance diagram of Figure 2. Elderly Parent represents John and is modeled as User. Elderly represents Joanna and is modeled as a Target Entity. Users will interact with the system through a mobile application in which it is modeled as a *Active Digital Artefact* which allows users to receive alerts anywhere (over the internet). In Figure 2 we have an overview of how users will interact with the system. Here we can see that a user can subscribe to the services of Notification (sending email, SMS) and Alarm Ring (alert with the noise in the cell phone) when an event is detected. These services are subscribed by the Caregiver Mobile APP which subscribes the services for detection of Invasion, Forgetfulness, Discomfort, and Leakage that are exposed from the JDL Based Data Fusion system modeled the Data Fusion Resource.

2) Identify the Environment that Target Entity will inhabit: The physical environments that the elderly inhabit is a house with bedroom, living room, kitchen and bathroom, and the virtual environments are social media like Twitter and Facebook. These environments are modeled in Figure 3.

3) Identify and select the contexts that Target Entity can participate: The Physical environments highlighted in Figure 3 enable the elderly to participate and perform the following contexts: water consumption, Electricity consumption, LPG consumption, and movement action. These contexts were modeled in the diagrams of the Figures 4, 5, 6 and 7. These contexts were selected because they represent dynamic variables and from them can be identified patterns and anomalies that will be important in the fusion process for the detection of the following events: Forgetting LPG consumption, Invasion of the house, Water leakage and Discomfort. The virtual environments described in Figure 3 and 5 enable the elderly to perform the following contexts: Twitter and Facebook Text Production. In LPG Consumption forgetfulness detection, Figure 4, the Movimentation, Water Consumption and Electricity Consumption are contexts that were used as "Context For" LPG Consumption and these relations were modeled as Ancillary Variable Rule. The LPG Consumption was used as "Context Of" forgetfulness and this relationship was modeled as Problem Variable Rule. In Discomfort Detection, Figure 5, the Twitter and Facebook Text Production and Electricity Consumption are contexts that were used as "Context Of" Discomfort and these relations were modeled as Problem Variable Rule. In Invasion detection, Figure 6, the Living Room Movimentation and Electricity Consumption are contexts that were used as "Context For" Bedroom Movimentation and these relations were modeled as Ancillary Variable Rule. The Bedroom Movimentation was used as "Context Of" Invasion and this relationship was modeled as Problem Variable Rule. In Water Leakage Detection, Figure 7, the Movimentation and Electricity Consumption are contexts that were used as "Context For" Water Consumption and these relations were modeled as Ancillary Variable Rule. The Water Consumption was used as "Context Of" Leakage and this relationship was modeled as Problem Variable Rule.

4) Identify the Resources: The main component for performing event detection is a data fusion sub-system based on the JDL model. This sub-system is modeled here as *Data Fusion Resource*. The association classes created from the context-event and context-context relationships shown in the Figures 4, 5, 6 and 7 define what will be the problem and ancillary variable in the data fusion process. The data flowing through the problem and ancillary variables are imported into the DFR by invoking the *Resources* that collects and provides access to them. The soft data produced in these contexts are collected by Twitter and Facebook NPL Applications modeled as Soft Data Capture Resource. The Hard data are exposed by GetSensorsValues modeled as On-Device Resource hosted in Micro-controller Board. These resources and their relationships are modeled in the diagram of Figure 8.

5) Map Resources to Device types: The system encompasses a set of devices endowed with sensors for movement, and water, gas, and electricity consumption. A micro-controller Board modeled as Device centralizes all data collected that will be required by the data fusion resource. They were modeled in the diagrams 3 and 2.

VI. Conclusion

A Domain Model in the context of system architecture plays a similar role as do ontologies in the context of semantic and knowledge sharing because it defines a point of reference for understanding a domain and agreeing on its definition, using some formalism [9]; in our case UML diagrams. Our domain model also provides a common lexicon and taxonomy of the PMS. The main purpose of a domain model is to generate a common understanding of the target domain in question. Only with a common understanding of the main concepts it becomes possible to argue about architectural solutions and to evaluate them. PMS domain model has concepts that allow us to design monitoring systems based on context-aware data fusion and deal with the personalization of monitoring and the unpredictability of events. The PMS domain model concepts were exemplified with the modelling of Elderly Personalized Monitoring System. The personalization of the monitoring was modeled according to the patterns of consumption and movement and the unpredictability of events was reduced by the detection of four distinct events using the same resources. In the future work we intend to make a framework for a Design Process Flow approach with an information model, resilience model, algorithms and ERD diagrams to assist with more artifacts the PMS design process.

References

[1] J. Gómez-Romero, M. A. Serrano, J. García, J. M. Molina, and G. Rogova, "Context-based multi-level information fusion for harbor surveillance," *Information Fusion*, vol. 21, pp. 173–186, 2015.

[2] H. Mshali, T. Lemlouma, M. Moloney, and D. Magoni, "A survey on health monitoring systems for health smart homes," *International Journal of Industrial Ergonomics*, vol. 66, pp. 26–56, 2018.

[3] U. N. H. S. Programme, U. N. H. S. P. R. O. for Asia, and the Pacific, *People's Process in Post-disaster, and Post-conflict Recovery and Reconstruction*. UN-Habitat Regional Office for Asia & the Pacific, UN-Habitat Fukuoka Office, 2008. [Online]. Available: https://books.google.com.br/books?id=A66tygAACAAJ

[4] L. Snidaro, J. García, J. Llinas, and E. Blasch, *Context-Enhanced Information Fusion: Boosting Real-World Performance with Domain Knowledge*, ser. Advances in Computer Vision and Pattern Recognition. Springer International Publishing, 2016. [Online]. Available: https://books.google.com.br/books?id=V2dBDAAAQBAJ

[5] Q. Ni, A. García Hernando, and I. de la Cruz, "The elderly's independent living in smart homes: A characterization of activities and sensing infrastructure survey to facilitate services development," *Sensors*, vol. 15, no. 5, pp. 11 312–11 362, 2015.

[6] L. Snidaro and I. Visentini, "Context for dynamic and multi-level fusion," pp. 431–451, 2016.

[7] M. M. Kokar, M. D. Bedworth, and C. B. Frankel, "Reference model for data fusion systems," in *Sensor Fusion: Architectures, Algorithms, and Applications IV*, vol. 4051. International Society for Optics and Photonics, 2000, pp. 191–203.

[8] C. Larman, *Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2004.

[9] S. Haller, A. Serbanati, M. Bauer, and F. Carrez, "A domain model for the internet of things," in *2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing*. IEEE, 2013, pp. 411–417.

[10] W. Elmenreich, "A review on system architectures for sensor fusion applications," in *IFIP International Workshop on Software Technolgies for Embedded and Ubiquitous Systems*. Springer, 2007, pp. 547–559.

[11] J. Llinas, L. Snidaro, J. García, and E. Blasch, "Context and fusion: Definitions, terminology," pp. 3–23, 2016.

[12] H. Carvalho, W. Heinzelman, A. Murphy, and C. Coelho, "A general data fusion architecture," in *Int. Conf. on Info. Fusion*, 2003, pp. 1465–1472.

[13] F. Bacha, K. Oliveira, and M. Abed, "Using context modeling and domain ontology in the design of personalized user interface," *International Journal on Computer Science and Information Systems (IJCSIS)*, vol. 6, pp. 69–94, 2011.

[14] J. García, L. Snidaro, and J. Llinas, "Architectural aspects for context exploitation in information fusion," pp. 185–203, 2016.

[15] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009. [Online]. Available: http://doi.acm.org/10.1145/1541880.1541882

[16] A. Bassi, M. Bauer, M. Fiedler, T. Kramp, R. van Kranenburg, S. Lange, and S. Meissner, *Enabling Things to Talk: Designing IoT Solutions with the IoT Architectural Reference Model*, 1st ed. Springer Publishing Company, Incorporated, 2016.