A DSL-based Approach for Detecting Activities of Daily Living by Means of the AGGIR Variables

José Manuel Negrete Ramírez, José Manuel Negrete Ramírez, Philippe Roose, Marc Dalmau, Yudith Cardinale, Edgar Silva

Article

A DSL-based Approach for Detecting Activities of Daily Living by Means of the AGGIR Variables

José Manuel Negrete Ramírez[1], Philippe Roose[2], Marc Dalmau[2], Yudith Cardinale[3] and Edgar Silva[3]

[1] LIFO, Institut National des Sciences Appliquées Centre Val de Loire, Université d'Orléans, Bourges, France, jose.negrete_ramirez@insa-cvl.fr;
[2] IUT de Bayonne - LIUPPA, Université de Pau et des Pays de l'Adour, Anglet, France; {Philippe.Roose,Marc.Dalmau}@iutbayonne.univ-pau.fr
[3] Dpto. de Computación y Tecnología de la Información; {ycardinale,11-10968}@usb.ve

Version July 13, 2021 submitted to Journal Not Specified

Abstract: In this paper, we propose a framework to study the AGGIR (Autonomie Gérontologique et Groupe Iso Ressources – Autonomy Gerontology Iso-Resources Groups) grid model, with the aim of assessing the level of independency of elderly people, in accordance with their capabilities of performing daily activities, as well as interacting with their environments. In order to model the Activities of Daily Living (ADL), we extend a previously proposed Domain Specific Language (DSL), by defining new operators to deal with constraints related to time and location of activities and event recognition. The proposed framework aims at providing an analysis tool regarding the performance of elderly/disabled people within a home environment by means of data recovered from sensors using a smart-home simulator environment. We perform an evaluation of our framework in several scenarios, considering five of the AGGIR variables (i.e., feeding, dressing, toileting, elimination, and transfers), as well as health-care devices for tracking occurrence of elderly activities. Results demonstrate the accuracy of the proposed framework to manage the tracked records correctly and thus generate the appropriate event information related to the ADL.

Keywords: Domain Specific Language; Feature-oriented Programming; Pervasive Computing; Pervasive Health Systems and Services; AGGIR grid.

1. Introduction

According the United Nations Department of Economic and Social Affairs (UN DESA)[1], population is aging faster than ever before. Due to this increase, reducing medical costs and improving quality of care service [1] have become in recent years a need for new care delivery mechanisms and structures [2]. Personal Sensor Networks (PSN) and Body Sensor Networks (BSN) in smart environments have become a viable alternative to traditional healthcare solutions. PSN are used to detect human daily activities and measure conditions within the environment. BSN are used to monitor vital signs and health conditions by measuring physiological parameters.

Several approaches propose different frameworks focusing on identifying the Activities of Daily Living (ADL) that require monitoring [3–12]. However, there is still a lack of ADL to be analysed, whereas others consider a subset of specific activities; but most importantly, they are not based on a specific tool such as the AGGIR (Autonomie Gérontologique et Groupe Iso Ressources – Autonomy

[1] https://www.un.org/development/desa/publications/world-population-prospects-the-2017-revision.html

Gerontology Iso-Resources Groups) grid [13]. The AGGIR grid is an autonomy assessment tool used in France to measure the indenpendncy level of elderly people.

In this context, in a previous work [14], we have presented a DSL which allows the representation of atomic activities composing the AGGIR grid in the form of a plot, providing a history file for detecting abnormal behavior of the inhabitants of the monitored house. Afterward, we developed a framework in which the DSL is integrated [15], with the aim to achieve the monitoring of a person within a smart home environment to identify the ADL performed by the inhabitants, by means of collection and analysis of data obtained from sensors located in the environment over a certain period of time.

In this paper, we extend our previous works by introducing the identification of complex activities and providing support for simulation and visualization. To do so, the DSL is extended by defining operators, which deal with constraints related to time, location, and event recognition. Moreover, this DSL is able to categorise a set of activities based on the constants of the AGGIR grid, which classifies autonomy levels to various environmental factors affecting the activities and social life of a person. Thus, more AGGIR activities can be detected, such as alimentation, dressing, toileting, elimination, and transfers. The framework is improved by adding support for more type of sensors and the capacity of performing configuration of the environments and visual simulations of the behaviour of people in such environments. Thus, the framework provides an analysis tool regarding the performance of elderly/disabled people within a home environment by means of data recovered from sensors using the iCASA simulator. In turn, the framework provides a general approach to detect ADL, relate them with the AGGIR variables, and determine the independence of elder people in their homes. To evaluate our approach, we pick five of the AGGIR variables (i.e., feeding, dressing, toileting, elimination, and transfers) and evaluate their testability in many scenarios, by means of records representing the occurrence of activities or unexpected behavior of the elderly. Furthermore, a health-care device use case concerning the employment of medical equipment for tracking blood glucose is presented. Results demonstrate the accuracy of our framework to manage the obtained records correctly and thus generate the appropriate event information.

The remainder of this paper is organized as follows. The AGGIR grid model as well as the definition and characteristics regarding different types of DSL are explained in Section 2. Related studies are presented in Section 3. The main features of the proposed DSL, as well as a brief review of operators regarding time, location, and event constraints is described in Section 4. Our framework proposal is explained in Section 5. The details of the experiments and the discussion about obtained results are given in Section 6. Finally, conclusions and future work are given in Section 7.

2. The AGGIR grid model and DSL

In this section, the AGGIR grid model, a tool introduced in January 2002 to expand senior benefits, is briefly described. Additionally, a review of the different types of DSL in any context (logic languages, textual languages, and graphical languages) is presented.

2.1. AGGIR Model

The AGGIR grid is an autonomy assessment tool used in France to measure the indenpendency level of elderly people. The AGGIR grid model is a six-level dependence scale (GIR1 to GIR6), that can be defined based on a set of seventeen three-state variables. Each variable can have one of these values: **A**, for complete dependency; **B**, for partial dependency; and **C**, for complete independency. The variables are classified into two groups: discriminatory and illustrative variables, as shown in Table 1.

Since DSL are recognized as effective tools to increase the productivity and quality of software development, in this work we propose a DSL in order to express the situations related to the AGGIR variables that respond to the ADL performed by people with a physical or mental disability, support for elderly, diseases connected to aging. The DSL allows to express situations related to the maintenance of people at home [16]. Subsequently, the definition and characteristics regarding several DSL categories are presented.

Table 1. AGGIR variables

Discriminatory Variables	Illustrative variables
Coherence	Management
Location	Cooking
Toileting	Housekeeping
Dressing	Transportation
Self-feeding / Alimentation	Purchases
Elimination	Medical treatment
Transfers	Leisure activities
Indoor movement	
Outdoor movement	
Distant communication	

2.2. Domain Specif Languages (DSL)

A DSL is a programming language, which helps developers to define concepts in terminology belonging to a particular domain [17–20]. DSL are conceived as small textual languages which are able to take certain narrow parts of programming and make them “easier to understand and therefore quicker to write, quicker to modify, and less likely to breed bugs”. Their main goal is to improve developers' productivity as well as reducing the communication gap in software development between programmers and domain experts [21]. Moreover, DSL are considered as increasingly popular software development techniques, which use concepts from the problem domain rather than the solution domain [22].

DSL are separated into two categories based on their design: internal DSL and external DSL [21]. An internal DSL (also called embedded DSL) provides a general-purpose host language to solve domain-specific problems (originally based on the LISP programming language) [23]. Due to the fact that an external DSL is independent from any other language, they need their own infrastructures like parsers, linkers, compilers, or interpreters [24]. External DSL represent 50% of the DSL [25].

Moreover, patterns [17] and design guidelines [26] have been proposed for the development of DSL. These guidelines are classified by: purpose, realization, content, concrete, and abstract syntax; as shown on Table 2. According to the concrete language, purpose, and domain; some guidelines might be contradicting or irrelevant [26], so they should be considered as proposals when designing a DSL [27]. The most common approach is to write requirements using natural language [28]. This can be used for different purposes to describe any kind of requirements and functional specifications for information systems [29].

3. State of the Art

Several DSL have been developed for different areas of application, e.g., expert rules, business rules, configuration rules. A systematic mapping study is presented in [25].

The conceptual DSL described in [30] expresses how business logic can be translated by means of automated code execution for the creation of logic-based smart contracts. Moreover, two execution modes for smart contracts are highlighted: lazy execution (initiated by an actor, either manually or scheduled); and eager execution (fully automated transactions).

IoTDSL, a prototype DSL meant to allow end users drive the IoT devices installed in their homes relying on a high-level rule-based language is presented in [31]; in order to achieve definition and manipulation of devices which are deployed in home environments. Users are able to describe and combine event-based semantics as well as structural configurations in a declarative manner; with a high-level representations of devices. The orchestration of events is then analysed to a component in charge of translating high-level rules into a complex event processing facility dedicated to evaluate

Table 2. Categories of DSL design guidelines proposed by Karsai *et al.* [26].

Category	Description	Guidelines
Language purpose	Goal of the DSL	1. Identify language uses early 2. Ask questions 3. Make your language consistent
Language realization	Implementation of the DSL	4. Decide carefully whether to use graphical or textual realization 5. Compose existing languages where possible 6. Reuse existing language definitions 7. Reuse existing type systems
Language content	Elements of the DSL	8. Reflect only the necessary domain concepts 9. Keep it simple 10. Avoid unnecessary generality 11. Limit the number of language elements 12. Avoid conceptual redundancy 13. Avoid inefficient language elements
Concrete syntax	Readable (external) representation of the DSL	14. Adopt existing notations domain experts use 15. Use descriptive notations 16. Make elements distinguishable 17. Use syntactic sugar appropriately 18. Permit comments 19. Provide organizational structures for models 20. Balance compactness and comprehensibility 21. Use the same style everywhere 22. Identify usage conventions
Abstract Syntax	Internal representation of the DSL	23. Align abstract and concrete syntax 24. Prefer layout which does not affect translation from concrete to abstract syntax 25. Enable modularity 26. Introduce interfaces

runtime events. Moreover, the proposed prototype relies upon a textual syntax; and simulation code can be generated to play configurations defined by the user.

A framework is proposed in [32] regarding complex event reasoning concerning process, integration, and provision of reasoning in order to address decision support to supply chain planners whenever information about disruption events (such as process failure) is incomplete or uncertain, by using logical and probabilistic reasoning approaches.

A DSL called ReqDL for describing requirements and capturing bidirectional traceability data, concerning system modeling elements is introduced in [28]. Additionally, a generation algorithm for independent trace models is presented; as well as concrete and abstract syntax in terms of grammar and metamodel. Also, three types of operators are proposed to describe requirements and capture trace data: attributes, description, model elements, and trace links. Moreover, a working example is also provided.

A graphical DSL provides domain experts with an intuitive and user-friendly tool to define the complex event processing domains of interest for which critical situations in real time need to be detected [33]. Some recent work on this research field are presented.

Dolphin, an extensible programming language implemented as a Groovy DSL for autonomous vehicle networks is described in [34] .It is designed to express an orchestrated execution of tasks defined for multiple vehicles dynamically available in a network. Additionally, the built-in operators include support for composing tasks in several forms, i.e., instance according to concurrent, sequential, or event-based task flow, partially inspired by process calculi approaches. Moreover, the integration of the aforementioned DSL with an open-source toolchain for autonomous vehicles is described, in which, users are able to edit and run Dolphin programs through a custom window, embedded in the overall GUI environment for editing and monitoring. Also, results from field tests using unmanned underwater vehicles and unmanned aerial vehicles are presented.

Authors in [35] introduce a DSL named SimulateIoT, a model-driven development approach aiming to define, generate code, and deploy the simulation of IoT systems, thus achieving their design as well as their deployment by means of a domain metamodel, a graphical concrete syntax, and a model to text transformation.

All these works demonstrate the utility of different types of DSL in several domains. With the purpose to support the identification of ADL performed at home by elderly/disabled people, many works have proposed the development of DSL. In the following section, we dedicate on the revision of existing DSL in the context of both description and recognition of ADL. Furthermore, a brief review of operators regarding time, location, and event constraints is introduced.

3.1. DSL for the detection of ADL

Authors in [36] introduce a smart city oriented infrastructure for collecting and managing data related to behavior patterns concerning elderly people and their daily activities, both in indoor and outdoor environments. Validation considering both low-level and high-level use cases was carried out. Part of the grammar employed regarding user motility and indoor-outdoor localization is presented: *BODY_STATE_START/ BODY_STATE_STOP* for indicating the change of a particular body state of the subject (i.e. still, walking, sleeping, etc.); *POI_ENTER/POI_EXIT* for managing the location type and/or the GPS coordinates. Furthermore, the user-environment interection is described by activities such as *FURNITURE_OPEN/ CLOSED* for sensing contact, motion and vibration on furniture and tools. Furthermore, *state_type*, *location_type* are defined in order to identify the performance of activities.

Authors in [37] propose a DSL for model-driven development of activity-oriented context-aware applications, in order to facilitate the development by improving the efficiency of developers. The concept model of such applications is analyzed, in order to design abstract syntax and concrete syntax. Implemented tools include the development environment as well as the generation of Java code and ontology specification. A case study and evaluation concerning a smart meeting room was introduced to demonstrate the utilities of the proposed approach.

An ambient assisted living architecture for the monitoring of elderly people is presented in [38]. The system is able to collect all information coming from the heterogeneous sensors located in the indoor environment, such as environmental parameters or biomedical information, and to forward it towards a remote service. The proposed system is able to continuously monitoring the elderly locomotor activity. Additionally, the system can trigger specific events when dangerous situations occur (e.g., fall detection). The feasibility of the proposed architecture was demonstrated by validation functional tests implementing a real supporting the elderly subject during his daily life activities.

Authors in [39] presented a study based on a model-driven solution for a top-down approach for rapid design and prototyping of an ambient assisted living capable of detecting the behaviour of elderly persons in their home, acquiring data through a sensor tag wristband that sends data to a smartphone application through bluetooth low energy protocol. The list of the detected low-level activities are: body state, indoor home monitoring, presence in indoor places, presence in outdoor places, smartphone usage, usage of home appliances, interaction with transportation and ambient parameters. Moreover, an example of grammar for low-level activities for the moving status concerning the *start* and *stop* moving actions .It includes *START_MOVING* and *STOP_MOVING*. Also, it is possible to calculate the duration of body state status along with the *STILL_TIME* value. Such a duration can be computed and treated as a measure.

A DSL for processing online events targeting binary sensors is presented in [40]. The approach is limited to handling binary sensors, due to the fact that the proposed operators are defined on the domain of boolean signals. The aforementioned language provides a primitive notion of state, modeled as a boolean signal over time, and allows the generation of complex conditions on different states using signal operators. Additionally, an interpreter for the language was implemented and applied it to rewrite a set of real ambient assisted living services.

A tool-based methodology, aimed to track ADL of elder adults, supporting replicable research is proposed in [41]; it allows processing sensor data with the purpose of defining a ruled-based monitoring process regarding the detection of the abovementioned activities. Moreover, it is intended to assist professional caregivers by providing functional awarenesses by means of a graphical tool; by taking advantage of user specific information and abstracting both complex and typical situations.

An approach with the objectif to evaluate a DSL focused on assistive services by professional caregivers is presented in [42]. Such an end-user language enables the formulation of assistive services by employing a domain-specific terminology. Also, features concerning the representation of circumstances where assitance needs to be provisioned are introduced; in addition to identifying the necessary interventions to be considered if assistance is required within context-aware systems. The DSL is only dedicated to the detection of contexts.

The research in [43] addresses a DSL conceived to support users with disabilities throughout the performance of a semi/automated cooking process. Furthermore, a graphical user interface (GUI) is furnished in order to access a set of instructions located within a cloud-based repository regarding meal preparation including microwave cooking directions. Additionally, it relies on a barcode scanner, a touchscreen, a set of speakers; as well as a set of environmental sensors. Audio and visual alerts regarding safety concerns are also provided.

Table 3 summarizes and evaluates the most recent DSL approaches regarding the services they provide, the categories of monitored daily activities, sensing types, and geriatric models.

Table 3. Comparison among the most recent DSL approaches regarding ADL's

Work	Services	Activities	Sensing	Sensing Type	Geriatric models
[39]	Movement tracking	IADL	Single	BSN	No
[37]	Activity-oriented context-aware applications, energy	ADL	Multi	PSN	No
[38]	Wellness determination, fall detection,mouvement tracking, energy	IADL	Multi	PSN, BSN	No
[36]	Detection of elderly behaviour patterns, mouvement tracking	ADL, IADL,	Multi	PSN, BSN	No
[40]	ADL estimation, energy	ADL	Multi	PSN	No
[41]	ADL estimation, energy	ADL	Multi	PSN	No
[42]	ADL estimation, energy	ADL	Multi	PSN	No
[43]	Assisting meal preparation	IADL	Multi	PSN	No

All these works consider the correct identification of the activities that require analysis of ADL. However, there is still a lack of ADL to be detected, while others only focus on a subset of specific activities; but most importantly, they are not based on a specific tool, such as the AGGIR grid.

3.2. Temporal, Localization, and Events Operators: A Review

In this section, we present a review of different operators used to describe time and location relations among events. Actions and activities that can be identified in a space are called events. These events can be tagged with time and location, that in turns make events be related according to the moment and location in which they occur.

According to study presented in [44], in order for a system to make sensible decisions, it has to be aware of where the users are and have been during some period of time. Spatial and Temporal logic is a well established area of Artifitial Intelligence (AI) [45], which has been applied to represent and reason on spatial and temporal features and constraints of context and situations [46]. For instances, temporal knowledge on human activities can be specified by means of the temporal operators `ANDlater` and `ANDsim` in Event-Condition-Action rules introduced in [47]. Authors in [48] apply Allen's Temporal Logic [49] (see Table 4) to describe, constrain, and reason on temporal sequences in dealing temporal and spatial knowledge in smart homes as well.

The event calculus of Shanahan [50], infers what is true from information that expresses what and when something occurs (actions) and what happens after those actions. Shanahan uses the action concept instead of event, but it makes no difference between the two. The notion of fluent is used to express anything that can change over time, as seen on Table 5.

Table 4. Allen's temporal operators

Relation	Symbol	Symbol for inverse	Illustration
X before Y	$<$	$>$ (after)	___X___ ___Y___
X equals Y	$=$	$=$ (equals)	___X___ ___Y___
X meets Y	m	mi (met by)	____X____ ____Y____
X overlaps Y	o	oi (overlapped by)	______X______ ______Y______
X during Y	s	di (contains)	___X___ _______Y_______
X starts Y	d	si (started by)	____X____ _______Y_______
X finishes Y	f	fi (finished by)	____X____ ________Y________

Table 5. Shanahan's event calculus formulas

Formula	**Meaning**
Initiates(α, β, τ)	Fluent β starts to hold after action α at time τ
Terminates(α, β, τ)	Fluent β ceases to hold after action α at time τ
InitiallyP(β):	Fluent β holds from time 0
$\tau_1 \leq \tau_2$	Time point τ_1 is before time point τ_2
Happens(α, τ)	Action α occurs at time τ
HoldsAt(β, τ)	Fluent β holds at time τ
Clipped(τ_1, β, τ_2)	Fluent β is terminated between times τ_1 and τ_2

Some others works worth revising are shown on Table 6, where temporal, spatial, and event-related operators are proposed in order to provide a significant contribution to a solution of these problems. We consider the temporal operators proposed by Allen [49], as well as operators such as "*Before, During, After, Begin, End, startTime, endTime*". With regard to location operators, we employ "*Inside, Outside, Joint, nextTo*", among others. We also utilise conventional mathematic operators such as "*greater than, lower than*", to indicate event conditions based on attributes.

Table 6. Spatial, temporal and event operators

	Temporal	Location	Event
Freksa [51]	☑	☐	☐
Ribarić and Dalbelo Bašić [52]	☑	☐	☐
Randell *et al.* [53]	☐	☑	☐
Galton [54]	☐	☑	☐
Randell *et al.* [55]	☐	☑	☐
Kim *et al.* [56]	☐	☑	☐
Pecora *et al.* [57]	☐	☑	☐
Bruno *et al.* [58]	☐	☑	☐
Patkos *et al.* [59]	☐	☑	☐
Santos *et al.* [60]	☐	☑	☐
Furze and Bennett [61]	☐	☑	☑
Angsuchotmetee *et al.* [62]	☐	☑	☑

4. Domain Specific Language

In a previous work we developed a DSL [14] in order to express situations related to the AGGIR variables that respond to the activities performed by people with a physical or mental disability to determine their independence at home. Afterward, we developed a framework [15], in which the DSL is integrated, intended to supervise an occupant within a smart home environment aiming to recognise several ADL carried out by the household residents over a certain period of time. In this paper, we extend the DSL by introducing the identification of complex activities. In this context, an atomic event is defined as an event that can be detected using one reading from a sensor; whereas a complex event is considered as an activity performed by an inhabitant that is detected by several sensors at the same time.

For carrying out such identification, sensors to identify the ADL are specified (Section 4.1 and Section 4.2); as well as the orchestration for atomic (Section 4.4.1) and complex events detection (Section 4.4.2). Moreover, the features that the extended DSL provides are the recovery of data from health related sensors and home-related sensors defined and described through the DSL in order to register operations regarding a specific date or a time range starting from a particular date or between two-time points, in addition to the identification of both atomic and complex events.

4.1. Smart Home Sensors

In the context of this work, the purpose of Wireless Sensor Networks (WSN) is to detect a user's vital signs, activities, and surroundings. To achieve this goal, a set of sensors is needed to measure vital signs from the inhabitant. Moreover, smart home sensors need to be deployed to monitor the surroundings in a home environment; so it is possible to identify the activities associated with the AGGIR constants, considering physiological contexts and environmental contexts.

Thus, for developing the proposed DSL, it is necessary to acknowledge which sensors must be employed for this matter. With regard to the identification of ADL by means of several sensors for providing non intrusive monitoring, a group of sensors that can describe such activities are proposed. The aforementioned sensors are enlisted in Table 7, where some examples of their use are proposed. Additionally data attributes and data types for each sensor are indicated.

Table 7. Smart Home Sensors

Sensor type	Attributes	Data type	Examples
Electromagnetic sensor	On/Off	`Boolean`	Cooker/stove; oven; light; switch
Proximity sensor	On/Off	`Double`	Sink
Capacitive sensor	On/Off	`Boolean`	Kitchen; counter; chair
Magnetic sensor	Open/Close	`Boolean`	Refrigerator door; cupboard doors
Presence Sensor	On/Off	`Boolean`	Room occupancy

Moreover, relevant data for each performed task identified through the data gathered by sensors, are considered in Table 8, in order to obtain information to achieve the identification of the AGGIR constants.

Table 8. Additional data for recognition of activities

Extra data type for each task	Data type
startTime	`Date / String`
endTime	`Date / String`
Duration	`Integer / Double`
Location	`String`
Day	`String / Integer`

4.2. Health Related Sensors

As previously mentioned, health care issues within the aging population represent a problem that needs to be addressed in the proposed DSL. Regarding this aim, a group of commercial medical sensors are suggested; such sensors are usually considered as non-intrusive, replaceable, and most of them are low cost. Medical sensors are mainly used in the medical field for the objective of pervasive healthcare. Some of the conventional medical sensors for physiology measurement are listed in Table 9. Moreover, each of the devices has specific requirements in terms of parameters, units and collected data.

Table 9. Medical sensors [63]

Device	Parameters	Units	Collected Data	
			Data Type	Range
Pulse and Oxygen in Blood (SPO2)	Pulse: 25-250 SPO2 : 35-100	ppm %	Integer Integer	25-250 ppm 35-100 %
Electrocardiogram (ECG)	Heart rate ECG signal	BPM (beats per minute) Volts	Integer Real	0-200 bpm 0/5 V
Airflow	Respiratory rate Breathing intensity	PPM (peaks per minute) Volts	Integer Real	0-60 PPM 0-3.3 V
Blood Pressure Sensor	Systolic pressure Diastolic pressure Pulse	mm Hg mm Hg ppm	Integer Integer Integer	0-300 mmHg 0-300 mmHg 30∼200 ppm
Glucometer	Glucose Glucose	mmol/L mg/dL	Real Real	
Body Temperature Sensor (BTS)	Body Temperature	Degree Celsius (°C)	Real	0-50 °C
Electromyography (EMG)	Muscle rate Muscle signal	CPM (contractions per minute) Volts	Integer Real	0-60 cpm 0-5 V
Spirometer (for breathing measuring)	Volume Air flow	l l/min	Real Integer	0.01∼9.99 L 50∼900 L/min
Galvanic skin response (GSR)	Conductance Resistance Voltage	Siemens Ohms Volts	Real Real Real	0-20 Siemens 10K-100KOhms 0-5V
Body Position	Body position Acceleration	Human body position G	String Real	5 positions
Snore	Snore rate Snore signal	SPM (Snores per minute) Volts	Integer Real	0-60 spm 0-5 V

4.3. Main characteristics of the proposed DSL

Due to the fact that understanding the process under which the proposed DSL works, it is relevant to highlight its most remarkable characteristics: (i) expressing situations related to the AGGIR variables that respond to the activities performed by the elderly/handicapped people at home; (ii) the representation of attribute-based event conditions; (iii) the representation of spatio-temporal event conditions. Those are defined by temporal operators, such as *Before, During, After.* To this end, the representation of the DSL realies on a GUI, in order to describe complex events.

DSL interface. We propose a graphical DSL rather than a textual language, to make it easy to handle, since the user might not be necessarily a person who is acknowledged in the programming field. Data recovery from sensors is crucial for the detection of ADL. For this matter, the proposed DSL plays an important role by representing the gathering of information related to the sensed environments; as well as returning graphical results by means of the aforementioned GUI. Additionally, the information present in the results can be saved as textual specifications in order to be reused, and thus facilitating to find the analysed data without performing any manipulation once again. This functionality allows its reusability by different context-aware applications, thus, not being limited only to the AGGIR variables.

DSL operations. The textual specifications generated by DSL GUI can be used to describe events from an operation perspective. Such as an operation is in charge of measuring one of the following options: a value, a maximum or minimum value, an average of a set of values, or graphically displaying measured values within a time domain. These operations are exposed as a programming interface: (i) the `Value` operation: which returns the value (s) of the device; (ii) the `Maximum` operation: which returns the maximum value of the value (s) of the device; (iii) the `Minimum` operation: that returns the minimum of the value (s) of the device; (iv) the `Average` operation: which returns the average of the value (s) of the device; and (v) the `Graphic` operation: that returns an icon with the value (s) of the device.

Due to the fact that events within a smart home environment are sensed during a specific time range, multiple possibilities to describe time using the DSL are considered. This includes determining a specific date or description of a time range starting from a particular date or between two-time points representing the beginning and end of the event. The specified time points can be either a date for a day or hours within a day. The `Value` operation can be calculated for the values: (i) on a specific date; (ii) between two hours; (iii) between two dates; (iv) from a specific date. The operations `Maximum`, `Minimum`, `Average`, `Graphic`, are be calculated for the values: (i) between two dates; (ii) from a date; and (iii) total (all available values for the device). After selecting the operation as well as the desired calculation means, it only remains to specify the date or the time according to what has been previously selected.

Since in the proposed DSL, the need to represent activities performed by the elderly or handicapped people within a home environment is present; it is imperative to be aware when such an activity takes place, as well as if there exists repetition or periodicity during the interval of its performance. By implementing such a methodology, the detection of anomalies on the behaviour of the inhabitants is intended, so coherence of the ADL can be ensured.

Spatio-temporal event conditions. In order to deal with temporal constraints, spatio-temporal event conditions are also represented by the DSL. Those are defined by temporal operators such as "*Before, During, After, Begin, End*", and those defined using spatial operators i.e., "*Inside, Outside, Joint*". The proposed criteria to determine the parameters for the example of the precedent paragraph, are: time and space, and within the time dimension. Also, it is relevant to consider:

i) Concurrency: for recognizing activities which take place simultaneously, but they do not necessarily require the user's interaction at the same time. That is, activities that have been started but not yet ended by the inhabitant [64].
ii) Precedence: for establishing a logical order of the activities, e.g., going to the bathroom and then washing hands.
iii) Simultaneity: for identifying which activity takes most of the time from the user, when multitasking capabilities might be present, e.g., preparing meal and calling on the phone; or watching television while eating.
iv) Recurrence: for determining a logical sequences of situations. In the case where there is a recurrence of an activity, it is essential to define at what this activity is carried out regularly.

Some examples of these four latter are presented on Fig. 1.

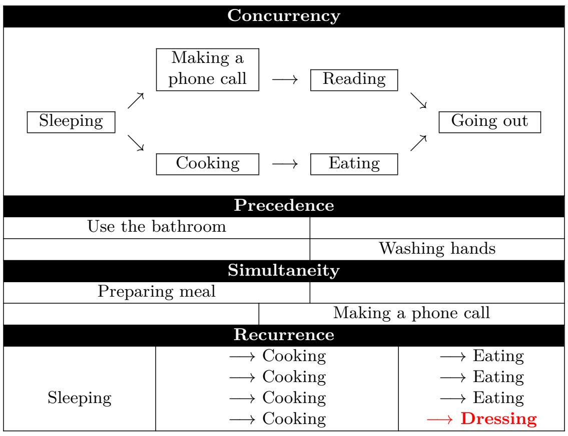

Figure 1. Criteria proposed over the time dimension

4.4. Representation of the AGGIR variables

So far, the main characteristics of the proposed DSL have been introduced. In this section, the explanation regarding the orchestration for both atomic and complex event detection is provided.

4.4.1. Atomic Event Detection

In order to recognize an atomic event, it is required to evaluate a given sensor reading against a predefined condition from a simple activity. In the proposed approach, an atomic event is simply defined as an ***Activity***, and each ***Activity*** represents one of different tasks composing every AGGIR variable. Table 10 shows how each ***Activity*** is associated to data obtained from sensors located within the smart-home environment concerning three of the AGGIR variables: toileting, dressing, and transfers. For the toileting AGGIR variable, three atomic events extracted from sensor readings are considered: the bathroom door sensor identifies the ***Activity*** regarding opening/closing the door; the toilet flush sensor indicates the ***Activity*** of activating the toilet flush; and the whashbasin proximity sensor describes the ***Activity*** of washing hands.

Table 10. Orchestration of activities

Elimination	Dressing	Transfers
Activity = $\sum(+)$ Sensor data		
Bathroom door sensor (*On–Off* / `Boolean`)	Wardrobe door sensor (*On–Off* / `Boolean`)	Bed capacitive sensor (*On–Off* / `Boolean`)
Toilet flush sensor (*On–Off* / `Boolean`)	Wardrobe proximity sensor (*On–Off* / `Double`)	Chair capacitive sensor (*On–Off* / `Boolean`)
Washbasin proximity sensor (*On–Off* / `Double`)		

4.4.2. Complex Event Detection

Subsequent to the description of atomic event detection, it is important to emphasize how complex event detection takes place in the proposed approach. For this subject, parting from readings from sensors defining a simple activity; a complex event can be considered as a sum of multiple events connected together representing each one of the AGGIR grid variables. That is, an event that summarizes, represents, or denotes a set of atomic events (Activities). In the proposed approach, a complex event is stated clearly as a ***Situation***.

In order to illustrate the composition of a ***Situation***, Table 11 indicates how three of the AGGIR variables are composed: (i) the toileting ***Situation*** is composed by the ***Activities***: *open bathroom door*, *use of toilet flush*, *wash hands*; (ii) the dressing ***Situation*** is composed by the ***Activities***: *open wardrobe door*, *spend time changing clothes*, *close wardrobe door*; and (iii) the Transfers ***Situation*** is composed by the ***Activities***: *lying down*, *sitting down*, *getting up*. Furthermore, concerning the achievement of each one of the aforementioned variables, the activities conforming such variables must be detected by following a specific sequence: one activity must be finished before the next one can be performed. That is, the precedence criteria must be considered in order for the variable to be completed; as pictured on Fig. 1.

Table 11. Orchestration of the AGGIR variables

AGGIR Variable = Situation		
Elimination	**Dressing**	**Transfers**
Situation = $\sum(+)$ Activities		
Open bathroom door	Open wardrobe door	Lying down
Use of toilet flush	Spend time changing clothes	Sitting down
Wash hands Close bathroom door	Close wardrobe door	Getting up

After describing the mechanism for orchestrating complex events; next section describes the proposed operators to represent complex events in the DSL.

4.4.3. Proposed operators

Allen's temporal operators. In order to deal with constraints regarding the time dimension, it is necessary to express conditions regarding temporal relations between sensor states, which are relevant for describing activities. To this end, the DSL takes advantage of the criteria over the time dimension precised on Section 4.3 and illustrated on Fig. 1: concurrency, precedence, simultaneity, recurrence. For example, the event outlining that shower is "on" during presence in the bathroom; detects an activity described between two sensors. For this purpose, a useful tool for representing temporal conditions between two time intervals are the Allen temporal relations [49] (Table 4).

With the aim to establish finite time intervals, temporal bounds to the Allen's temporal relations were applied; e.g. the constraint X *MEETS* $[\mathrm{t}_1, \mathrm{t}_2][0, \infty)$ Y implies that event Y should be *met by* event X, that the start time of X must occur between t_1 and t_2; and that the end time of X should occur right before the beginning of Y.

Figures 2 to 6 present an example of how temporal conditions can be used to model activity recognition in our approach, alongside with location based operators; which are presented in the next paragraphs.

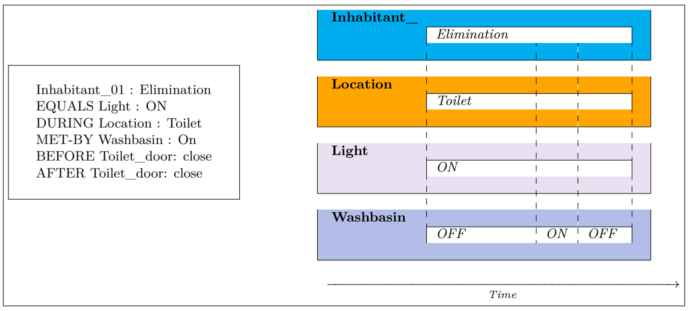

Figure 2. Elimination

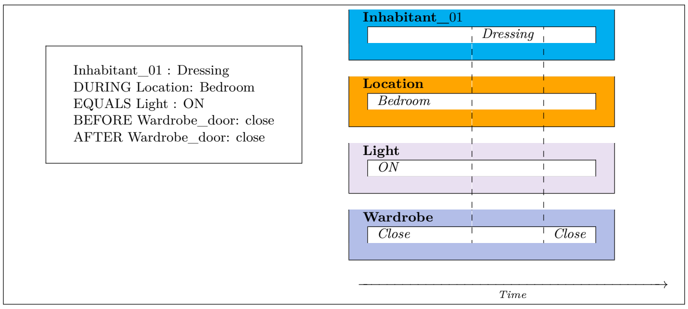

Figure 3. Dressing

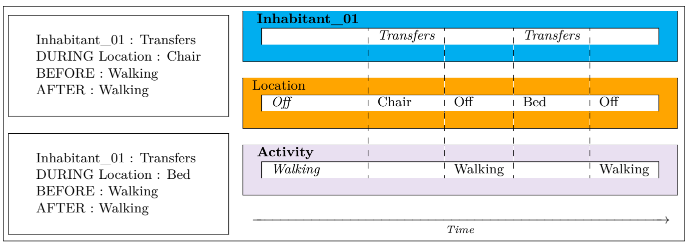

Figure 4. Transfers

Location based operators. Such operators are in charge of helping determine the position of the inhabitant within a home environment; as well as the description of ADL through sensor readings. To this effect, the proposed approach considers location based operators: "*Inside*", "*Outside*" and "*Joint*"; in order to locate in space the ADL performed for the inhabitant. The description regarding each one of the proposed location based operators are presented on Table 12.

Event based operators. Additionally, with regard to determination of planned tasks that should be carried out by the inhabitant, such as taking medications; event based operators are important for this matter. Those operators are presented on Table 13.

Next section is dedicated to the presentation of the framework proposal for the validation and experimentation of the DSL.

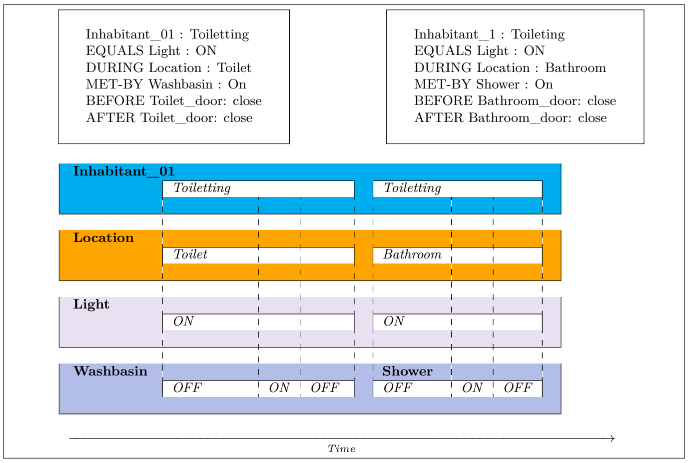

Figure 5. Toileting

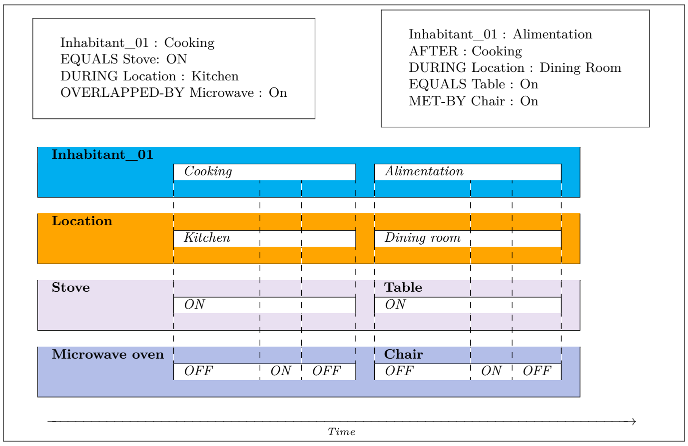

Figure 6. Alimentation

Table 12. Location based operators

Operator	Description	Data type
Inside	Returns true if the user is inside a given area.	String
Outside	Returns true if the user is outside a given area.	String
Joint	Returns true if the user is in two locations at the same time	String

Table 13. Event based operators

Operator	Description	Data type
Planned task (medical measurements)	Returns true if the user performs a planned task	String
Unplanned tasks (medical measurements)	Returns true if the user performs an unplanned task	String

5. A Framework to Evaluate the AGGIR Variables: Our Approach

Subsequently to the proposed DSL, a framework is introduced to fulfill the requirements for processing complex events regarding the AGGIR grid model generated by data recovered from sensors within a smart home environment; in order to evaluate the level of independency of elderly people, according to their capabilities of performing activities and interact with their environments over the time.

5.1. Framework Modules

The main purpose of the presented framework is to encourage users who are not necessarily acknowledged in the programming field to be able to define events according to the AGGIR grid variables; by providing a high-level abstraction, which makes it easy and intuitive by means of concepts that are close to the final users.

The proposed framework is composed by three main modules, as shown on Fig. 7: (i) the simulator module; (ii) the descriptor module; and (iii) the analyzer module. All of them are supported by the DSL. A brief description of each one is provided in the next paragraphs.

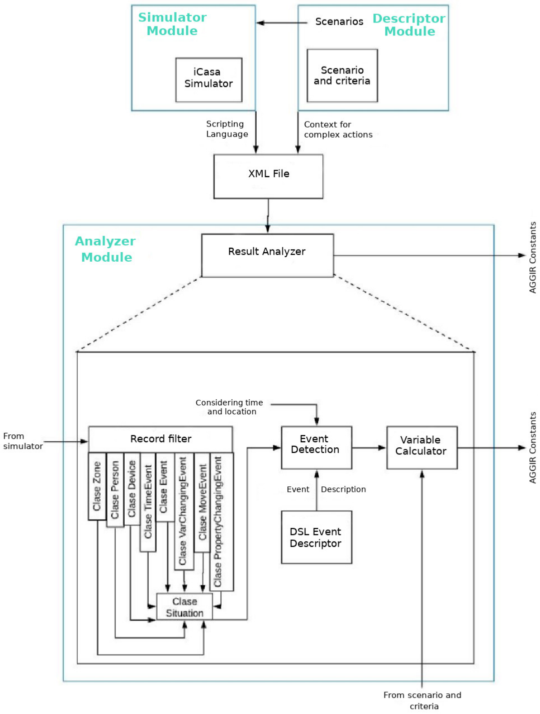

Figure 7. General Architecture of the proposed framework

Simulator module: In order to describe the elderly's ADL, smart-home environment scenarios need to be parametrized. For this purpose, the activities performed by the subject are carried out and information is recovered from sensors located within the smart home environment. As the Simulator module, iCASA [65] was integrated to the proposed framework, which is a smart home simulator which allows to have control over: time, environment, inhabitants, devices, a GUI, scripting facilities (for the environment), and notification facilities [66], as enlisted on Table 14. iCASA is used in order to set up a simulated scenario.

Table 14. iCASA facilities

Time	Possibility to slow down, speed up or stop time during the simulation. Simulation of long-term actions such as energy consumption to skip to important actions.
Environment	Definition of different zones in a house. An administration interface to modify different physical properties (temperature, luminosity, etc.) of the different zones, is provided.
Inhabitants	Insertion or removal inhabitants from the environment. Inhabitants, can move from zone to zone, may be carrying physical devices.
Devices	Devices can be simulated or real. At any time, the user can add or remove new simulated devices and modify their localisation in the rooms.
A graphical user interface	The interface displays a map of the house and the localisation of the different devices. It allows to create and configure devices, create and move physical users and watch their actual configurations.
Scripting facilities	Support to the scripts writing to control the environment. Scripts provide a convenient way to test the applications under reproducible conditions.
Notifications facilities	iCASA is event-based and is able to notify subscribers of any modifications in the environment.

Descriptor module: For the purpose of helping users to interact with the framework, the Descriptor module provides a GUI, in order to describe all the criteria over the time dimension, location, and events, as well as the required scenarios for the simulation. Also, sensors to identify the ADL have to be specified through the GUI. A group of sensors that can describe such activities are introduced on Table 7.

Moreover, with regard to the scenario specification concerning a certain period of time, the descriptor module is in charge of the definition of the parameters with respect to location of both ADL and sensors, type of sensors, time, and events that need to be identified: (i) Location Map, which is applied to refer to the representation of the environment by means of the house plan where the implementation of the sensor network takes place; (ii) Sensor network, that is focused on defining information related to the sensor network environment infrastructure; to this end, relevant data should be managed, such as: the inventory of sensors located within the smart-home environment, as well as the location where each sensor is implemented; (iii) Sensor reading, due to the fact that information retrieved from sensors are in a raw format, such data need to be organized for the purpose of easing their retrieval and interpretation during the event detection process; (iv) Event condition, since conditions are established for triggering the detection of events, such conditions must be defined by the user; and (v) Event occurrence, once the event conditions have been provided, the event occurrence must be managed in accordance with such aforementioned conditions.

Analyzer module: In order to recover data from the Simulator module and the Descriptor module; the Analyzer module is proposed to carry out such a task. Once all the necessary data is collected, the analyzer module analyzes them in order to classify them and evaluate if the AGGIR variables of the case study have been carried out to completion. The Analyzer module consists of: (i) a record filter; (ii) an event detector; and (iii) a variable calculator. Such components are described hereafter.

Record Filter: Once the simulation has been conducted, in order to manage the obtained data, the record filter of the Analyzer module separates the recover data into a series of records. Each record is responsible of the description of actions collected by the sensor network located within the smart-home environment. Therefore, each register is conformed by data files, i.e., the time when the action takes place (corresponding to the simulator clock), the sensor ID, and the sensed value. Furthermore, because of the retrieval of such values is generated in raw data format, it is necessary to translate them by means of the Analyzer module with the aim of achieving the identification of events, as well as calculation of values related to the AGGIR grid variables.

With the aim of performing a filtering of all entries, the first task of the analyzer module is to classify such inputs in a correct manner. For this matter, the generation of “filters” was modeled in the present approach as classes. These latter were introduced in order to determine the type of elements that integrate the analyzed set of instructions, by giving each element the appropriate attributes for the modeling of raw data. To this extent, the proposed generated classes for data classification are enlisted in Table 15. Subsequently, in order to detect the performed activities, the *Event detector* is presented.

Table 15. Classes for data classification

Class name	Function
Zone	Modelling the simulator zones; in charge of identifiers, measurements, variables and their values.
Device	Managing several aspects concerning the simulator devices; i.e. identifiers, properties, values spatial location.
Person	Controlling the inhabitants within the smart-home environment; such as the person identifiers and their location with respect to the simulation context.
TimeEvent	Modelling the instructions concerning the time dimension: generating real time data parting from time raw data, allowing time management during the analysis.
Event	Considered as a basic event; modelling situations not requiring additional data.
MoveEvent	Modelling events regarding movement within a specific zone in the smart-home area, either a person or a device. Subclass of the *Event* class.
VarChangingEvent	Modelling events concerning modifications to designated variables to a specific zone, i.e. temperature change. Subclass of the *Event* class.
PropertyChangingEvent	Modelling events that generate modifications with regard to intrinsic properties regarding the simulation devices.
Situation	Modelling complex situations. Complex situations are a result of the interaction of several atomic events which allows to identify abnormal behaviour

Event detector: After all relevant data from PSN and BSN have been gathered, the identification of events must be carried out. To this extent, the Event detector relies on the introduced DSL to achieve such identification, by managing information grouped in multiple records which are extracted from several sensors

Aiming to separate each reading detected by several devices, the Event detector retrieves information in terms of the scripting language. Such events are identified as low-level actions instances; e.g., if a presence sensor is off in room X, and after a period of time another presence sensor is on in room Y; it can be inferred that the action "walk" was carried out.

Regarding the recognition of malfunctions, the detection of abnormal situations is carried out at the level of identification of events and situations. For this matter, throughout the analysis of low-level events (i.e., Situation class instances) collected from device readings from the simulator module script, it was possible to extract a set of anomalies to be detected by the aforementioned module. Such anomalies are enlisted hereafter in Table 16.

Table 16. Anomalies identified by the event detector module

Sensor/Activity	Detected problem	Issue for the inhabitant
FloodSensor	Leak or break in pipelines	Flood within the home environment
DimmerLight / BinaryLight	Exceeded MAX time ON	High energy cost; neglecting of ADL's; location, coherence.
	ON at a wrong time	
Heater/Cooler	Exceeded MAX time ON	High cost, neglect; abnormal thermal sensation; location problem
	is ON when no needed	Neglecting of household tasks, location, coherence
Not leaving room	Exceeded MAX time ON	Housekeeping, alimentation, elimination, leisure activities.
Light/Presence	Inhabitant stays still	Accident in ZONE, location, coherent behaviour, transfer.
CO/CO_2 sensor	High CO/CO_2 concentration	Household tasks; i.e. impossibility to check on the status on pipelines
Door sensor	Door *LEFT OPEN*	Location, coherence, household maintaining.
Siren	Resident must leave the house	Maintenance of home.
Toilet sensors: Door toilet flush, washbasin	Irregular urination time.	Toileting, elimination, personal hygiene, tranfers.
Wardrobe door	Not changing clothes	Dressing, coherence, toileting
Motion sensor	Wandering around at unusual time	Location, coherence
Temperature/Motion	Abandoning kitchen while cooking	Location, coherence, household tasks.

Each one of the aforementioned anomalies is related to the AGGIR grid variables. The detection of any of these issues may return a negative value for the associated variable, that is to say, the inability to satisfactorily carry out the AGGIR variable in question.

Variable calculator: Regarding the achievement of the AGGIR variables, criteria consisting on accomplishing a determined number of events during a specific time lapse related to the evaluated AGGIR variable must be fulfilled; e.g., in order to validate if the dressing AGGIR variable is consummated, AGGIR dressing event conditions such as moving towards the wardrobe at least twice a day must be met. For this purpose, although every AGGIR variable is based on three major states, which identify that the elderly inhabitant possesses the ability to perform the ADL conforming a specific variable, whether it is completely, partially, or not existent at all; this study covers a proposal which only relies on two cases out of the three abovementioned options; meaning that either the inhabitant is in complete possession of the skills concerning the performance of the ADL composing the evaluated AGGIR variable or simply not.

5.2. Complex Event Detection

Provided the main components of the aforementioned framework proposal, the description of the process for achieving the detection of complex events is then referred.

Due to the fact that ADL take place within a domestic surroundings, sensors for detecting such ADL are placed within a smart-home environment. For this matter, such an environment is defined and configured through the Simulator module. Then, the detection process is triggered by the readings obtained from the sensors located within the smart home environment simulator.

In order to obtain information to achieve the identification of the AGGIR variables after every activity has been carried out to completion, some relevant data have to be considered, e.g., *startTime*, *endTime*; as shown in Table 8. All these data are specified in the Descriptor module. After the simulation took place, a history log file is created, which considers the logical aspect of situations, i.e., if the inhabitant is eating and its noon, it might seem logical, but the fact that the inhabitant is eating in the toilet is not logical.

As a result, the history log file is necessary to deduct the possible activities which will be performed in the future, and to find a relation among them to assure a coherent behaviour. Such a file will permit to obtain information whether it is from long or short periods of time; making possible the identification of complex situations inside the home environment, such as feeding, toileting, transfers; where data collected through the timeline of activities is useful to determine if the behaviour of the inhabitant can be considered as normal.

6. Experimental Evaluation: Use Case Description

In order to demonstrate the efficacy and suitability of our proposed approach, a set of experiments with regard to the detection of the AGGIR grid variables are performed. For the purpose of evaluating the introduced DSL, four of the AGGIR variables (i.e., dressing, toileting, transfers, and feeding) are picked with the aim of determining their testability in many scenarios, by means of records representing the occurrence of ADL performed the elderly inhabitant within a one-week period. To detect either the achievement or absence of the AGGIR variables by means of the DSL, we follow a methodological process for the experimental simulations, consisting on the following steps:

Step 1. In order to prepare the scenario for an elderly indoor daily routine over the course of one week, the first step consists on the specification of the the criteria over the time and location of activities, type and location of sensors, and events that have to be detected; in this specific case, all activities related to the dressing, toileting, feeding, and transfer AGGIR variables. For this matter, the aforementioned actions are specified and described through the GUI of the Descriptor module. To this end, the primary source of information used to generate the scenario is the schedule proposed by the work of [9], where the behaviour of an inhabitant living in a genuine household environment is simulated based on the daily routine of an elderly resident, as shown in Fig. 8. However, in order to make the scenario more suitable for the simulation, many modifications take place. Furthermore, for the purpose of making the scenario more suitable for the simulation, the XML format is used to describe the scenario programmatically, given that such files allow to define events chronologically; thus facilitating the processing carried out by the Simulator module. A brief example of the above-mentioned files provided in Fig. 9.

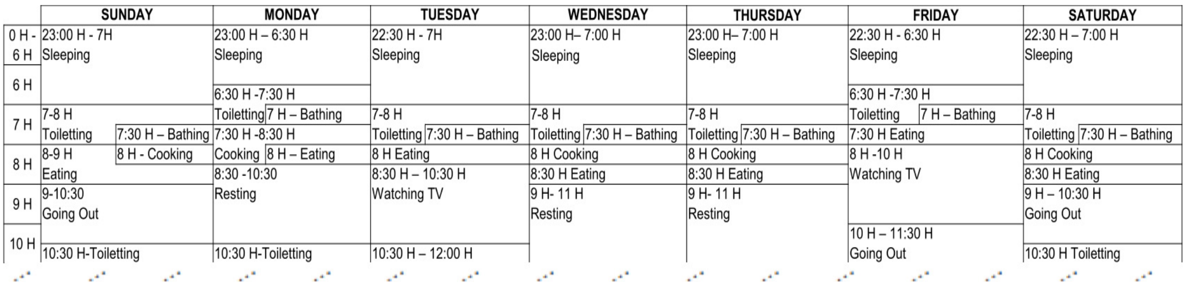

Figure 8. Schedule employed in the simulation

Step 2. Once the information provided by the Descriptor module is established; the Simulator module, i.e., the iCASA framework, executes the simulation according to such precised information.

Step 3. After the simulation is performed, the record filter component of the Analyzer module organizes the resulting data into a set of records, each of which represents an action captured by a sensor within the smart home environment (Fig. 7). To this end, each record consists of data fields, such as the time it occurred (according to the simulator clock), the sensor ID, and the sensed value. Moreover, due to the fact that such values are raw data, they must be interpreted by the Analyzer module in order to generate the detection of events and then calculate the values of the AGGIR variables. Subsequently, having collected all the results, the next step is the detection of events. For this purpose, an event is considered as a composite act that is described by data provided by several sensors; that is to say, an

XML File Syntax	
Attribute	**Description**
startdate	Starting date of script execution
factor	Speed factor for execution

```
<behavior startdate="27/06/2017-00:00:00" factor="1440">
```

Zone creation	
Attribute	**Description**
id	Zone's Identifier
leftX	Left X coordinate value
topY	Top Y coordinate value
bottomZ	Bottom Z coordinate value
X-Length	Zone X-Length
Y-Length	Zone Y-Length
Z-Length	Zone Z-Length

```
...
  <create-zone id="bedroom"  leftX="55" topY="370" X-Length="259" Y-Length="210" />
  <create-zone id="bathroom"  leftX="55" topY="20" X-Length="260" Y-Length="350" />
  <create-zone id="Toilet"  leftX="143" topY="299" X-Length="45" Y-Length="45" />
```

Device creation	
Attribute	**Description**
id	Device's Identifier
type	Device's type

```
...
  <create-device id="Pres-A1010" type="iCasa.PresenceSensor" />
  <create-device id="Pres-A1011" type="iCasa.PresenceSensor" />
  <create-device id="Pres-A1012" type="iCasa.PresenceSensor" />
```

Person displacement	
Attribute	**Description**
personId	Person's Identifier
zoneId	Zone's identifier

```
...
  <move-device-zone deviceId="Pres-A1009" zoneId="Chairs" />
  <move-device-zone deviceId="Pres-A1010" zoneId="Oven" />
  <move-device-zone deviceId="Pres-A1011" zoneId="washingbasin" />
```

Person creation	
Attribute	**Description**
id	Person's Identifier
type	Person's type

```
...
  <create-person id="Paul" type="Grandfather" />
```

Person displacement	
Attribute	**Description**
personId	Person's Identifier
zoneId	Zone's identifier

```
...
  <move-person-zone personId="Paul" zoneId="bedroom"  />
  <move-person-zone personId="Paul" zoneId="bathroom"  />
  <move-person-zone personId="Paul" zoneId="Toilet"  />
```

Simulation delay	
Attribute	**Description**
value	Time in (virtual) units
unit	Time unit, value it can be h (hours), m (minutes, default), s (seconds)

```
...
  <move-person-zone personId="Paul" zoneId="Sofa"  />
  <delay value="1" unit="h"/>
  <delay value="30" unit="m"/>

...
```

Figure 9. Excerpt of an XML format file scenario employed by the simulator module

event relies on more than one record in raw data. Afterwards, the DSL is in charge of providing an accurate and unified description of the events. Once the event detection has occurred, the aim of the variable calculator within the Analyzer module is indicating whether an AGGIR variable has been accomplished or not.

In order to test the whole approach, we show its application to several scenarios, as we present in the following section.

6.1. Experimental Results

We designed two scenarios for the experimental evaluation. First scenario, consisting on an ideal week case scenario, in which all the ADL related to a specific AGGIR variable performed by the inhabitant are conducted successfully. Second scenario, with regard to the generation of a failure on the daily routine of the inhabitant, by randomly dropping some of the ADL conforming a specific AGGIR variable. Moreover, in order to generate the required simulations, the aforementioned scenarios consist on two sets of inputs managed by the proposed framework. Each one of such scenarios is conformed by seven days (one-week period).

In what follows, the main characteristics of the simulation scenarios concerning the identification of ADL as well as the AGGIR grid variables are described. In addition, since it is required to evaluate the ability of the proposed approach to deal with larger time domains, as well as several records and events; the scalability of the introduced framework is also illustrated.

Toileting, Dressing, and Transfer AGGIR Variables

Conditions regarding the achievement of a certain number of events within a period of time must be accomplished for the evaluation of the AGGIR variables. In order to detect the toileting AGGIR variable the following conditions must be met: the inhabitant must use the toilet, and wash his hands after eating or using the toilet, for at least three times a day. To calculate the transfer variable, namely the ability of a person to perform the basic movements of his daily routine, such as rising from bed, sitting down, and standing up from a chair; it is considered that there must be at least three sitting events a day, either taking place in the living room or in the kitchen, and at least rising from the bed one time per day. While the verification of the dressing variable relies on dressing events such as approaching the wardrobe at least twice a day.

In this sense, we represent all these variables as follows. The conditions for the achievement of the toileting AGGIR variable are: (i) the inhabitant must use the toilet; (ii) after, the chain of the toilet flush must be pulled; (iii) after using the toilet, the inhabitant should have his hands washed. In order to analyze such a variable, the records created by the presence sensor located within the bathroom area, alongside with the records issued from the use of both the toilet flush sensor and the washbasin proximity sensor, must be examined. To identify the AGGIR dressing variable, the rules for the detection of ADL composing such a variable are: (i) the inhabitant must get close to the wardrobe area; (ii) the inhabitant must spend time changing clothes. To this end, data originated by the wardrobe door sensor, as well as the wardrobe proximity sensor, must be gathered. For the purpose of recognizing the AGGIR transfer variable, the constituent events regarding the aforementioned variable must occur: (i) getting up from bed; (ii) taking a seat; (iii) standing up from a chair; not necessarily in that order. In this matter, capacitive sensors located in both bed and chair/armchair are responsible for the data collection.

In order to illustrate the three AGGIR variables outlined above, Fig. 10 provides a description of all activities related to the three considered AGGIR variables, as well as sensors related to each one.

With a view to perform the evaluation of the AGGIR variables, all the simulating and processing operations are performed by means of two sets of inputs for the introduced framework, in order to generate two different scenarios. For this matter, the week is separated into seven days, each of which is represented by means of one simulation file within the smart home environment simulator.

AGGIR Variable = Situation		
Toileting	**Dressing**	**Transfers**
Situation = $\sum(+)$ Activities		
Open bathroom door	Open wardrobe door	Lying down
Use of toilet flush	Spend time changing clothes	Sitting down
Wash hands Close bathroom door	Close wardrobe door	Getting up
Activity = $\sum(+)$ Sensor data		
Bathroom door sensor (*On–Off* / `Boolean`)	Wardrobe door sensor (*On–Off* / `Boolean`)	Bed capacitive sensor (*On–Off* / `Boolean`)
Toilet flush sensor (*On–Off* / `Boolean`)	Wardrobe proximity sensor (*On–Off* / `Double`)	Chair capacitive sensor (*On–Off* / `Boolean`)
Washbasin proximity sensor (*On–Off* / `Double`)		

Figure 10. Orchestration of activities / AGGIR Variables

To this effect, the first scenario is simulated of an ideal week case scenario, meaning that all the ADL were performed with no impediment by the inhabitant throughout the entire week, by means of a simulated sensor network within a house environment for monitoring the ADL carried out by the elderly resident. Results obtained from the simulation are presented in Table 17, in which, the number of detected activities per day related to each one of the considered AGGIR variables are shown. The check mark means that the criteria for determining that the person is independent (i.e., minimum number of activities related to each variable) has been reached.

For the purpose of generating a malfunctioning on the developed criteria, some of the ADL related to the evaluated AGGIR variables were randomly excluded in the second scenario during the seven-day period of analysis. Once the simulation was conducted, the obtained results demonstrate that the detection of complex events regarding either the achievement or the absence of ADL were carried out successfully, according to the pre-established conditions, as enlisted on Table 18.

Table 17. Results of the first scenario.

		Transfer	Toileting	Dressing
No. of events	MON	☑ 7	☑ 8	☑ 3
	TUE	☑ 6	☑ 8	☑ 3
	WED	☑ 7	☑ 11	☑ 2
	THU	☑ 7	☑ 10	☑ 2
	FRI	☑ 8	☑ 8	☑ 3
	SAT	☑ 8	☑ 6	☑ 4
	SUN	☑ 6	☑ 9	☑ 3
	Total	☑ 49	☑ 59	☑ 19

Table 18. Results of the second scenario.

		Transfer	Toileting	Dressing
No. of events	MON	☑ 7	☑ 7	☑ 3
	TUE	☑ 7	☑ 7	☒ 0
	WED	☑ 7	☑ 11	☑ 2
	THU	☑ 6	☑ 11	☑ 3
	FRI	☑ 8	☑ 8	☑ 3
	SAT	☒ 3	☑ 6	☑ 4
	SUN	☑ 6	☒ 6	☑ 2
	Total	☑ 44	☒ 56	☑ 16

Furthermore, the proposed method succeeded in identifying three simulated problems within a week. The days were randomly chosen. Fig. 11 and Fig. 12 show that despite the number of records in the second scenario related to personal transfer and toileting did not change significantly compared to the first analyzed scenario, however a problem with the dressing variable was detected, as illustrated on Fig. 13. This, in turn, reflects how the DSL event descriptor can perform a smart analysis of events.

AGGIR alimentation variable

With the aim of evaluating the alimentation AGGIR variable, for the first scenario, the ADL related to cooking must be performed by the inhabitant before the alimentation AGGIR variable takes place. That is to say, the accomplishment of preconditioned events conforming the cooking activity is necessary; more particularly: (i) open/close the refrigerator door; (ii) open/close the kitchen cabinet door; (iii) use of the stove/burners; (iv) use of the traditional/microwave oven; not necessarily in that order. Whereas, as for the second scenario, some of the abovementioned events concerning the cooking task were dropped. For this matter, information collected from sensors located within the kitchen area must be analyzed. To this end, data originated by both the refrigerator and kitchen cabinet door sensors, the stove magnetic sensor, as well as the oven / microwave oven sensor, must be gathered.

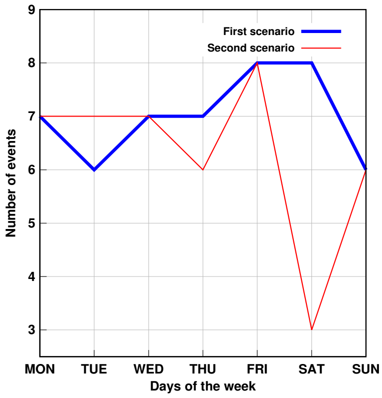

Figure 11. AGGIR Transfer events

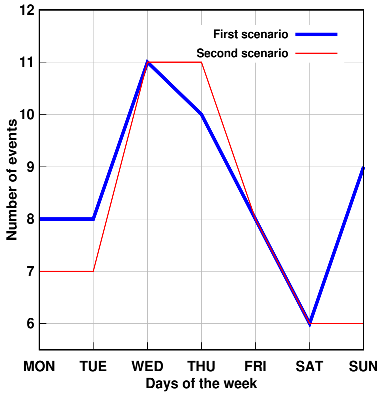

Figure 12. AGGIR Toileting events

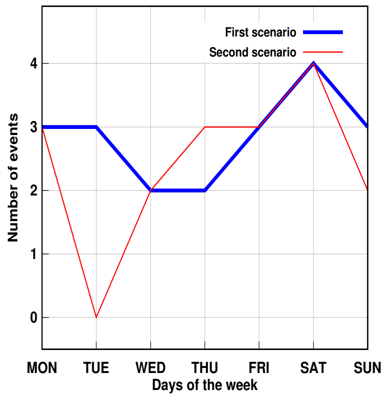

Figure 13. AGGIR Dressing events

Nonetheless, it is important to remark that according to the AGGIR grid, cooking is not conceived as a discriminatory variable, neither as an evaluation criterion for the alimentation AGGIR variable; which only requires the elderly being capable of serving and eating a prepared meal. For this reason, in our approach, cooking is considered as an ADL which can trigger the Alimentation AGGIR variable, not meaning to contradict the pre-established evaluation criteria, which can also be used for detecting the aforementioned variable in this work.

Once the cooking task is achieved, additionally to the first scenario, events defining the alimentation AGGIR variable must be identified, such as: (i) sitting within the dining room area; (ii) placing meal on the table; (iii) spending time consuming food. Moreover, with the aim of identifying the occurrence of the abovementioned events, data form sensors located within the dining room area must be collected, such as: chair capacitive sensor, table capacitive sensor, dinning room presence sensor. With respect to the second scenario, some of the ADL conforming the alimentation AGGIR variable, were arbitrarily left out. Additionally, in order to assure a coherent self-feeding behaviour, both scenarios must occur at least three times a day. The orchestration of activities regarding the alimentation AGGIR variable is illustrated on Fig. 14.

Cooking
Location: Kitchen
$\sum(+)$Activities
Open refrigerator door
Open cabinet door
Use of stove
Use of oven
$\sum(+)$Sensor data
Refrigerator door sensor (*On-Off* / `Boolean`)
Cabinet door sensor (*On-Off* / `Boolean`)
Stove electromagnetic sensor (*On-Off* / `Boolean`)
Oven electromagnetic sensor (*On-Off* / `Boolean`)

After –>

AGGIR Variable = Situation
Alimentation
Location: Dining room
Situation=$\sum(+)$Activities
Sitting down
Placing meal on table
Spending time eating
Activity=$\sum(+)$Sensor data
Chair capacitive sensor (*On-Off* / `Boolean`)
Table capacitive sensor (*On-Off* / `Boolean`)
Dining room presence sensor (*On-Off* / `Boolean`)

Figure 14. Orchestration of the AGGIR Alimentation Variable

Having completed the simulations of both scenarios, results concerning the first scenario proves that, indeed, the detection of complex events conforming the cooking tasks, as well as those from the alimentation AGGIR variable were carried out to completion successfully, in accord with to the predetermined input conditions, as enlisted on Table 19, in which, check marks indicate that the minimum number of detected events related to the aforementioned variable have been performed satisfactorily.

Moreover, as for the second scenario, due to the fact that some of the activities were not taken in account for the analysis of both the cooking activity and the alimentation AGGIR variable; the proposed framework also succeded in identifying either the concerned ADL was performed or was not completed according to initial parameters, as presented on Table 20.

Table 19. Results of the first scenario

		Cooking	Alimentation
No. of events	MON	☑ 12	☑ 7
	TUE	☑ 11	☑ 9
	WED	☑ 9	☑ 9
	THU	☑ 11	☑ 8
	FRI	☑ 10	☑ 8
	SAT	☑ 10	☑ 9
	SUN	☑ 9	☑ 7
	Total	☑ 72	☑ 57

Table 20. Results of the second scenario

		Cooking	Alimentation
No. of events	MON	☒ 7	☑ 9
	TUE	☑ 10	☒ 6
	WED	☑ 11	☑ 8
	THU	☒ 5	☑ 8
	FRI	☑ 9	☒ 3
	SAT	☒ 7	☒ 5
	SUN	☑ 10	☑ 9
	Total	☒ 59	☒ 48

Furthermore, in order to illustrate the number of events related to the cooking task, as well as for the alimentation AGGIR variable, Fig. 15 and Fig. 16 introduce by means of a graphical representation the events detected for both scenarios. Additionally, it can be observed for each case that the second scenario present a significant threshold with respect to the first scenario in some of the simulation weekdays, which implies that both the cooking activity and the alimentation AGGIR variable were not achieved particularly on those days; and hence, during the evaluated week concerning the second scenario.

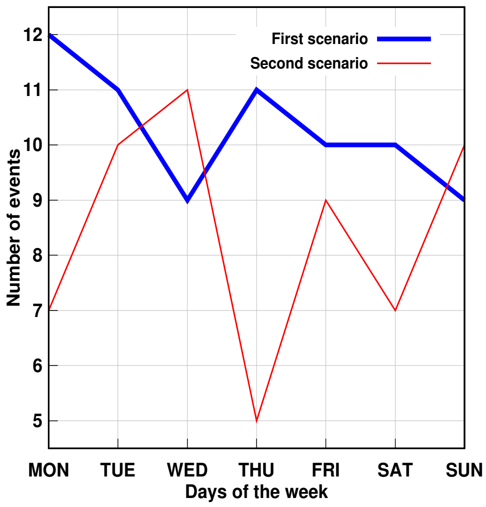

Figure 15. Cooking events

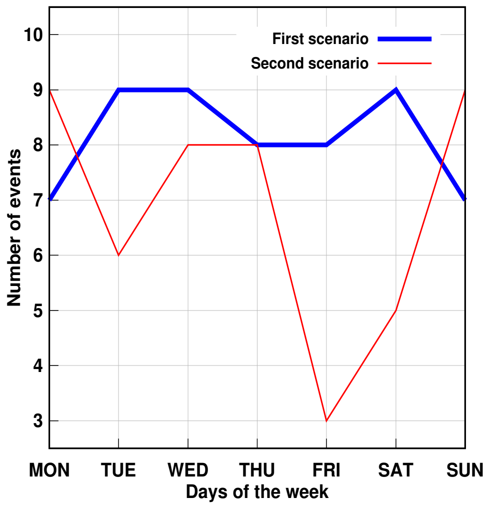

Figure 16. AGGIR Alimentation events

Health-care device use case

In order to prove the flexibility of the proposed framework, a health-care device use case is introduced. This case considers a resident in charge of taking glucose level measurements by means of a glucometer, performing a type of measurement other than those listed in the AGGIR grid. We design two scenarios, in which to ensure continuous measuring, it is necessary for the house resident to perform such measurements at least three times a day. To this extent, the first simulation scenario consists of a week during which the person was completely responsible of employing a glucometer device for measuring his own glucose levels, meaning that the completion of a planned medical task was achieved. The second scenario was thought to randomly exclude some of the measurements throughout several days of the week. All the simulations and operations are carried out by the proposed framework by means of records corresponding to each scenario. Results obtained from the simulations corresponding to both scenarios, are presented in Table 21.

With regard to the simulation results, it can be observed in Fig. 17, that the events occurred during the simulation week for the first scenario were satisfactorily detected, which implies that the measurement of the glucose level were achieved in accordance with the preset criteria, meaning that the house resident is in charge of planned tasks in a coherent manner. Additionally, information obtained from the second scenario shows that the inhabitant did not succeed in taking charge of a planned task, due to the fact that the number of identified events per day were not performed coherently (three to six times a day); more precisely, tuesday, thursday, and sunday.

Table 21. Results of the first scenario

		First scenario	Second scenario
No. of events	MON	☑ 3	☑ 6
	TUE	☑ 3	☒ 1
	WED	☑ 6	☑ 3
	THU	☑ 4	☒ 0
	FRI	☑ 3	☑ 3
	SAT	☑ 3	☑ 4
	SUN	☑ 5	☒ 2
	Total	☑ 27	☒ 19

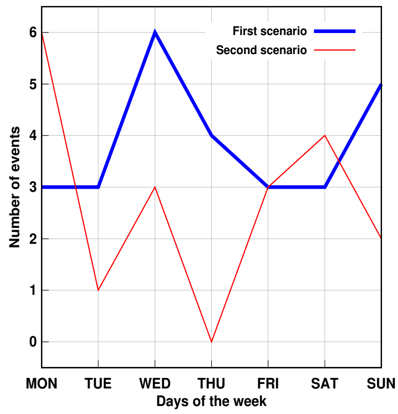

Figure 17. Glucose device events

This latter can be interpreted as the efficiency of the framework to perform the recognition of planned tasks concerning measurements carried out by the user by means of medical devices.

We also perform experiments to evaluate the framework scalability and performance In what follows, we describe the results regarding these experiments.

6.2. Framework scalability

To validate the execution of the framework under event processing conditions regarding both larger time domains and higher workload cases, the scalability of the framework for a period of time longer than one week is tested. For this matter, a time lapse of three months was simulated in order to generate the corresponding records. Next, the proposed criteria were applied with the intent to create the events, and thus calculate the values of three AGGIR variables: toileting, transfers, and dressing. In addition, with a view to allow flexibility when processing records and generating events, the assumption that each month is conformed by thirty days was considered. By doing so, there is a possibility to perform the required simulations by employing an approximate number of days per month; i.e., 28, 29, 30, and 31 days (Fig. 18).

Finally, to make sure that the algorithm is scalable, the elapsed time during the calculation of the three abovementioned AGGIR variables for simulation time periods for one day, one week, two weeks, three weeks, a month, two months, and three months; were monitored. Table 22 summarizes the elapsed time for each case. The values displayed include the time spent for each simulation, as well as results analysis, events generation, and finally the calculation of the AGGIR variable.

Table 22. Real time elapsed to calculate the constants in seven different simulation time ranges (***D***ay, ***W***eek, ***M***onth).

Time period	$1D$	$1W$	$2W$	$3W$	$1M$	$2M$	$3M$
No. of days	1	7	14	21	30	60	90
Time (s)	62	434	867	1301	1863	3713	5590

In order to present the results thus achieved, Fig. 19 shows the performance of the algorithm with simulating scenarios carried out with different time ranges. As a result, the obtained diagram is linear, which corresponds to the complexity of the algorithm of the Analyzer module, represented by $O(n)$, where n defines the number of events, implying that the framework as a whole, is scalable.

With the aim of applying the above-mentioned steps, several use cases are provided; namely, the alimentation, toileting, cooking and transfer AGGIR variables; as well as a description regarding an use

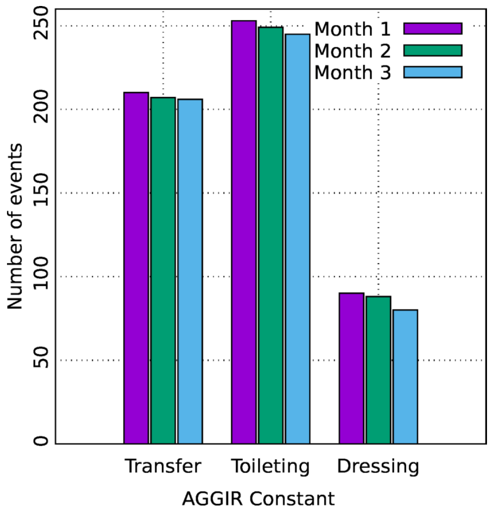

Figure 18. Three-month algorithm scalability

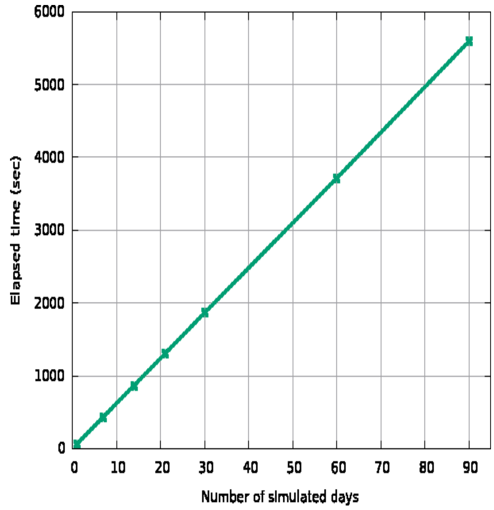

Figure 19. Analysis of Algorithm performance

case of a medical device for measuring blood glucose level. Moreover, results obtained show that the proposed framework suceeding on detecting the events and therefore the AGGIR variables according to preestablished conditions present in the experimentation scenarios.

7. Conclusions

In recent years, it has been reported that the population sector targeted as elderly is increasing more rapidly than in the previous decades. Besides that, age-related health issues as well as different chronic diseases start to become apparent at this stage of life. Moreover, various surveys indicate that elderly people are willing to live independently in their own homes as long as possible. In order to assess the abovementioned problem, PSNs and BSNs in smart environments represent an option regarding healthcare solutions for monitoring elderly people at home. However, despite their popularity, the absence of a monitoring approach with regard to a dependency evaluation model such as the AGGIR grid variables has never been treated. In this context, the core contributions of this research work are: (i) a state-of-the-art related to the proposed subject and future circumstances with regard to aging population, as well as the capabilities of technology for assisting the maintenance and monitoring the health needs of people, are reviewed; (ii) a DSL relying on the AGGIR grid evaluation model for assessing the performance of ADL of the elderly/handicapped at home; (iii) a framework tool for validating the proposed DSL in terms of its capacity of processing complex events; this framework offers a smart-home simulator called iCASA for carrying out evaluation and experimentation and a parser in charge of processing the data issued from the DSL aimed to create the appropriate instructions for the iCASA platform.

We are currently working on proposing the identification of a three-state variable approach for the AGGIR grid. Due to fact that the current version of the introduced DSL is only able to recognise two of the three values (**A** for complete dependency and **C** for complete independency) of the AGGIR grid variables, the proposal of an improved version of the DSL comprehending the three states of identification (including **B** for partial dependency), is one of the possible directions of the future work. For the purpose of assessing AGGIR variables that can be considered as more elaborate, such as coherence, further experimentation is required to evaluate crucial activities describing the concerned variable. Even though the presented approach considers the evaluation of the AGGIR discriminatory variables; with the goal of covering the whole spectrum of the AGGIR variables, the integration of the AGGIR illustrative variables is one of the next steps to be taken. So far, the perspective introduced in this research relies on the AGGIR grid variables. In order to expand the coverage of the suggested approach, the inclusion of other dependency evaluation models (e.g., the Functional Autonomy Measurement

System – SMAF model), represents a field of opportunity to broaden the scope of the present work. Besides the presence of the house residents, visitors can represent a factor for change regarding the household dynamics. For this matter, the adaptability of the proposed approach must be determined by tests concerning unknown users to the smart-home environment; as well as the impact that their visit may cause to the actual inhabitants. Despite the fact of having employed a smart-home simulator, as well as a schedule describing the actual activities that were carried out by the user; the application of the present work for real-time use cases is recommended. To this extent, monitoring of ADL within an actual smart-home environment would be ideal for testing the present approach in a real-time scenario.

Author Contributions: J.-M. Negrete Ramirez and Y. Cardinale conceived and designed the experiments: Research for analyzing the proposed model, analyzing the experiments and revising the proposed model; E. Silva performed the experiments; P. Roose, M. Dalmau and Y. Cardinale supervised the project; J.-M. Negrete Ramirez and Y. Cardinale wrote the paper.

Acknowledgments: The authors would like to acknowledge the the Institut National des Sciences Appliquées Centre Val de Loire and the Laboratoire d'Informatique Fondamentale d'Orléans for supporting the publication of the research herein presented.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ADL	Activities of Daily Living
AGGIR	Autonomy Gerontology Iso-Resources Groups
AI	Artifitial Intelligence
BPM	Beats per minute
BSN	Body Sensor Networks
BTS	Body Temperature Sensor
CPM	Contractions per minute
DSL	Domain Specific Language
ECG	Electrocardiogram
EMG	Electromyography
GSR	Galvanic skin response
GUI	Graphical User Interface
IoT	Internet of things
PPM	Peaks per minute
PSN	Personal Sensor Networks
SMAF	Functional Autonomy Measurement System
SPM	Snores per minute

References

1. Varshney, U. Pervasive healthcare. *Computer* **2003**, *36*, 138–140.
2. Kamel Boulos, M.N.; Lou, R.C.; Anastasiou, A.; Nugent, C.D.; Alexandersson, J.; Zimmermann, G.; Cortes, U.; Casas, R. Connectivity for healthcare and well-being management: examples from six European projects. *Int. J. of environmental research and public health* **2009**, *6*, 1947–1971.
3. Jih, W.r.; Hsu, J.Y.j.; Wu, C.L.; Liao, C.F.; Cheng, S.y. A multi-agent service framework for context-aware elder care. In AAMAS, 2006.
4. Lu, C.H.; Fu, L.C. Robust location-aware activity recognition using wireless sensor network in an attentive home. *IEEE Trans. on Automation Science and Engineering* **2009**, *6*, 598–609.
5. Lemlouma, T.; Laborie, S.; Roose, P. Toward a context-aware and automatic evaluation of elderly dependency in smart homes and cities. Int. Symp. on World of Wireless, Mobile and Multimedia Networks. IEEE, 2013, pp. 1–6.
6. Suryadevara, N.K.; Mukhopadhyay, S.C. Wireless sensor network based home monitoring system for wellness determination of elderly. *IEEE Sensor Journal* **2012**, *12*, 1965–1972.

7. Shimokawara, E.; Kaneko, T.; Yamaguchi, T.; Mizukawa, M.; Matsuhira, N. Estimation of basic activities of daily living using zigbee 3d accelerometer sensor network. ICBAKE. IEEE, 2013, pp. 251–256.
8. Chernbumroong, S.; Cang, S.; Atkins, A.; Yu, H. Elderly activities recognition and classification for applications in assisted living. *Expert Systems with Applications* **2013**, *40*, 1662–1674.
9. Yuan, B.; Herbert, J. Context-aware hybrid reasoning framework for pervasive healthcare. *Personal and ubiquitous computing* **2014**, *18*, 865–881.
10. Hooda, D.; Rani, R. Ontology driven human activity recognition in heterogeneous sensor measurements. *Journal of Ambient Intelligence and Humanized Computing* **2020**, *11*, 5947–5960.
11. Tsanousa, A.; Meditskos, G.; Vrochidis, S.; Angelis, L. A novel feature selection method based on comparison of correlations for human activity recognition problems. *Journal of Ambient Intelligence and Humanized Computing* **2020**, *11*, 5961–5975.
12. Gulati, N.; Kaur, P.D. FriendCare-AAL: a robust social IoT based alert generation system for ambient assisted living. *Journal of Ambient Intelligence and Humanized Computing* **2021**, pp. 1–28.
13. Dupourqué, E.; Schoonveld, S.; Bushey, J.B. AGGIR, the Work of Grids. *Long-term Care News* **2012**, *32*, 1–11.
14. Negrete Ramírez, J.M.; Roose, P.; Dalmau, M.; Bakni, M. Proposal and Validation of a Domaine Specific Language for the Representation of the AGGIR Constants. HEALTHINF, 2018, pp. 438–445.
15. Negrete Ramírez, J.M.; Roose, P.; Dalmau, M.; Cardinale, Y. An event detection framework for the representation of the AGGIR variables. 2018 14th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). IEEE, 2018, pp. 153–160.
16. Negrete Ramírez, J.M.; Roose, P.; Dalmau, M. Distributed interfaces and context-oriented broadcast services in a smart-home environment. Int. Conf. on Wireless and Mobile Computing, Networking and Communications. IEEE, 2016, pp. 1–8.
17. Mernik, M.; Heering, J.; Sloane, A.M. When and how to develop domain-specific languages. *ACM computing surveys (CSUR)* **2005**, *37*, 316–344.
18. Van Deursen, A.; Klint, P. Domain-specific language design requires feature descriptions. *Journal of Computing and Information Technology* **2002**, *10*, 1–17.
19. Sprinkle, J.; Mernik, M.; Tolvanen, J.P.; Spinellis, D. Guest editors' introduction: What kinds of nails need a domain-specific hammer? *IEEE software* **2009**, *26*, 15–18.
20. Shen, L.; Chen, X.; Liu, R.; Wang, H.; Ji, G. Domain-Specific Language Techniques for Visual Computing: A Comprehensive Study. *Archives of Computational Methods in Engineering* **2021**, *28*, 3113–3134.
21. Fowler, M. *Domain-specific languages*; Pearson Education, 2010.
22. Goulão, M.; Amaral, V.; Mernik, M. Quality in model-driven engineering: a tertiary study. *Software Quality Journal* **2016**, *24*, 601–633.
23. Henderson, P. Functional geometry. Proceedings of the 1982 ACM symposium on LISP and functional programming. ACM, 1982, pp. 179–187.
24. Efftinge, S.; Eysholdt, M.; Köhnlein, J.; Zarnekow, S.; von Massow, R.; Hasselbring, W.; Hanus, M. Xbase: implementing domain-specific languages for Java. ACM SIGPLAN Notices. ACM, 2012, Vol. 48, pp. 112–121.
25. Kosar, T.; Bohra, S.; Mernik, M. Domain-specific languages: A systematic mapping study. *Information and Software Technology* **2016**, *71*, 77–91.
26. Karsai, G.; Krahn, H.; Pinkernell, C.; Rumpe, B.; Schindler, M.; Völkel, S. Design guidelines for domain specific languages. *arXiv preprint arXiv:1409.2378* **2014**.
27. Vogel, T. Model-driven engineering of self-adaptive software. PhD thesis, Universität Potsdam, 2018.
28. Haidrar, S.; Anwar, A.; Bruel, J.M.; Roudies, O. A Domain-Specific Language to manage Requirements Traceability. *JSW* **2018**, *13*, 460–480.
29. Pfeiffer, M.; Pichler, J. A comparison of tool support for textual domain-specific languages. Proceedings of the 8th OOPSLA Workshop on Domain-Specific Modeling, 2008, pp. 1–7.
30. de Kruijff, J.; Weigand, H. An introduction to Commitment Based Smart Contracts using ReactionRuleML. VMBO, 2018, pp. 149–157.

31. Amrani, M.; Gilson, F.; Englebert, V. Complex Event Processing for User-Centric Management of IoT Systems. International Conference on Model-Driven Engineering and Software Development. Springer, 2017, pp. 426–448.
32. Nawaz, F.; Janjua, N.K.; Hussain, O.K. PERCEPTUS: Predictive complex event processing and reasoning for IoT-enabled supply chain. *Knowledge-Based Systems* **2019**.
33. Boubeta-Puig, J.; Ortiz, G.; Medina-Bulo, I. ModeL4CEP: Graphical domain-specific modeling languages for CEP domains and event patterns. *Expert Systems with Applications* **2015**, *42*, 8095–8110.
34. Lima, K.; Marques, E.R.; Pinto, J.; Sousa, J.B. Dolphin: a task orchestration language for autonomous vehicle networks. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018, pp. 603–610.
35. Barriga, J.A.; Clemente, P.J.; Sosa-Sánchez, E.; Prieto, Á.E. SimulateIoT: Domain Specific Language to design, code generation and execute IoT simulation environments. *IEEE Access* **2021**.
36. Mulero, R.; Almeida, A.; Azkune, G.; Abril-Jiménez, P.; Waldmeyer, M.T.A.; Castrillo, M.P.; Patrono, L.; Rametta, P.; Sergi, I. An IoT-aware approach for elderly-friendly cities. *IEEE Access* **2018**, *6*, 7941–7957.
37. Li, X.S.; Tao, X.P.; Song, W.; Dong, K. AocML: A Domain-Specific Language for Model-Driven Development of Activity-Oriented Context-Aware Applications. *Journal of Computer Science and Technology* **2018**, *33*, 900–917.
38. Mainetti, L.; Manco, L.; Patrono, L.; Secco, A.; Sergi, I.; Vergallo, R. An ambient assisted living system for elderly assistance applications. 2016 IEEE 27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC). IEEE, 2016, pp. 1–6.
39. Caione, A.; Fiore, A.; Mainetti, L.; Manco, L.; Vergallo, R. Top-Down Delivery of IoT-based Applications for Seniors Behavior Change Capturing Exploiting a Model-Driven Approach. *Journal of Communications Software and Systems* **2018**, *14*. doi:10.24138/jcomss.v14i1.438.
40. Volanschi, N.; Serpette, B.; Carteron, A.; Consel, C. A Language for Online State Processing of Binary Sensors, Applied to Ambient Assisted Living. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* **2018**, *2*, 192.
41. Belloum, R.; Consel, C.; Volanschi, N. A tool-based methodology for long-term activity monitoring. Proceedings of the 13th ACM International Conference on PErvasive Technologies Related to Assistive Environments, 2020, pp. 1–10.
42. Giraud, S.; Volanschi, N.; Consel, C. Empowering Caregivers To Customizing The Assistive Computing Support of Older Adults-An End-User Domain-Specific Approach. *International Journal of Human–Computer Interaction* **2020**, *36*, 1447–1459.
43. Zallio, M.; Kelly, P.; Cryan, B.; Berry, D. A co-Design approach to develop a smart cooking appliance. Applying a Domain Specific Language for a community supported appliance. *arXiv e-prints* **2021**, pp. arXiv–2101.
44. Cook, D.J.; Augusto, J.C.; Jakkula, V.R. Ambient intelligence: Technologies, applications, and opportunities. *Pervasive and Mobile Computing* **2009**, *5*, 277–298.
45. Liao, L.; Fox, D.; Kautz, H. Extracting places and activities from gps traces using hierarchical conditional random fields. *Int. J. of Robotics Research* **2007**, *26*, 119–134.
46. Ye, J.; Dobson, S.; McKeever, S. Situation identification techniques in pervasive computing: A review. *Pervasive and mobile computing* **2012**, *8*, 36–66.
47. Augusto, J.C.; Liu, J.; McCullagh, P.; Wang, H.; Yang, J.B. Management of uncertainty and spatio-temporal aspects for monitoring and diagnosis in a smart home. *Int. J. of Computational Intelligence Systems* **2008**, *1*, 361–378.
48. Gottfried, B.; Guesgen, H.W.; Hübner, S. Spatiotemporal reasoning for smart homes. In *Designing Smart Homes*; Springer, 2006; pp. 16–34.
49. Allen, J.F. Maintaining knowledge about temporal intervals. In *Readings in qualitative reasoning about physical systems*; Elsevier, 1990; pp. 361–372.
50. Shanahan, M. The event calculus explained. In *Artificial intelligence today*; Springer, 1999; pp. 409–430.
51. Freksa, C. Temporal reasoning based on semi-intervals. *Artificial intelligence* **1992**, *54*, 199–227.
52. Ribarić, S.; Dalbelo Bašić, B. Modelling Crisp and Fuzzy Qualitative Temporal Relations. *Journal of Information and Organizational Sciences* **2001**, *25*, 81–91.

53. Randell, D.A.; Cui, Z.; Cohn, A.G. A spatial logic based on regions and connection. **1992**.
54. Galton, A. Lines of sight. AISB Workshop on Spatial and Spatio-Temporal Reasoning, 1994.
55. Randell, D.; Witkowski, M.; Shanahan, M. From images to bodies: Modelling and exploiting spatial occlusion and motion parallax. IJCAI, 2001, pp. 57–66.
56. Kim, J.; Choi, H.s.; Wang, H.; Agoulmine, N.; Deerv, M.J.; Hong, J.W.K. POSTECH's U-Health Smart Home for elderly monitoring and support. Int. Symp on World of Wireless Mobile and Multimedia Networks. IEEE, 2010, pp. 1–6.
57. Pecora, F.; Cirillo, M.; Dell'Osa, F.; Ullberg, J.; Saffiotti, A. A constraint-based approach for proactive, context-aware human support. *J. of Ambient Intelligence and Smart Environments* **2012**, *4*, 347–367.
58. Bruno, B.; Grosinger, J.; Mastrogiovanni, F.; Pecora, F.; Saffiotti, A.; Sathyakeerthy, S.; Sgorbissa, A. Multi-modal sensing for human activity recognition. Int. Symp. on Robot and Human Interactive Communication. IEEE, 2015, pp. 594–600.
59. Patkos, T.; Plexousakis, D.; Chibani, A.; Amirat, Y. An event calculus production rule system for reasoning in dynamic and uncertain domains. *Theory and Practice of Logic Programming* **2016**, *16*, 325–352.
60. Santos, P.E.; Martins, M.F.; Fenelon, V.; Cozman, F.G.; Dee, H.M. Probabilistic self-localisation on a qualitative map based on occlusions. *J. of Experimental & Theoretical Artificial Intelligence* **2016**, *28*, 781–799.
61. Furze, T.A.; Bennett, B. Using the Principles of Classical Conditioning to Learn Event Sequences. *Computational Models Of Cognitive Development* **2011**, pp. 40–47.
62. Angsuchotmetee, C.; Chbeir, R.; Cardinale, Y.; Yokoyama, S. A dynamic event detection framework for multimedia sensor networks. Asia-Pacific Conference on Communications. IEEE, 2017, pp. 1–6.
63. S.L., L.M.S. MySignals - eHealth and Medical IoT Development Platform. Technical Guide. http://www.libelium.com/downloads/documentation/mysignals_technical_guide.pdf, 2019.
64. Helaoui, R.; Niepert, M.; Stuckenschmidt, H. Recognizing interleaved and concurrent activities: A statistical-relational approach. 2011 IEEE International Conference on Pervasive Computing and Communications (PerCom). IEEE, 2011, pp. 1–9.
65. Lalanda, P.; Hamon, C.; Escoffier, C.; Leveque, T. iCasa, a development and simulation environment for pervasive home applications. Consumer Communications and Networking Conference. IEEE, 2014, pp. 1142–1143.
66. Lalanda, P.; McCann, J.A.; Diaconescu, A. *Autonomic computing*; Springer, 2013.