

# CAPAS: A context-aware system architecture for physical activities monitoring

Paulo Ferreira<sup>1</sup>, Leandro O. Freitas<sup>1</sup>, Pedro Rangel Henriques<sup>1</sup>, Paulo Novais<sup>1</sup> and Juan Pavón<sup>2</sup>

<sup>1</sup>*ALGORITMI Centre, University of Minho, Braga, Portugal.*

<sup>2</sup>*Facultad de Informática, University Complutense Madrid, 28040 Madrid, Spain.*

*{pauloaferreira, leanfrts}@gmail.com, {prh, pjon}@di.uminho.pt, jpavon@fdi.ucm.es*

**Keywords:** Attribute Grammar, Intelligent Environment, Activity Recognition.

**Abstract:** Attribute grammars are widely used by compiler-generators since it allows complete specifications of static semantics. They can also be applied to other fields of research, for instance, to human activities recognition. This paper aims to present CAPAS, a Context-aware system Architecture to monitor Physical Activities. One of the components that is present in the architecture is the attribute grammar which is filled after the prediction is made according to the data gathered from the user through the sensors. According to some predefined rules, the physical activity is validated after an analysis on the attribute grammar, if it meets those requirements. Besides that it proposes an attribute grammar itself which should be able to be incorporated in a system in order to validate the performed physical activity.

## 1 Introduction

Nowadays it is a well known fact that practicing physical activities can prevent a lot of diseases both in the short and long term. However, getting injured is relatively easy if the activity is not being executed in the right way. So, activity monitoring systems are of the vital importance.

Attribute Grammars are such an important part of a program once it defines its semantic. Semantic is what gives a program meaning. Attribute grammars were first introduced in (Knuth, 1968) allowing compilers to process programming languages in a much easier way, since the parts from the compiler which dealt with semantics were becoming very complex considering the evolution of programming languages and computers.

Besides the consolidation of it in programming languages it is possible to apply this type of formalism to any kind of application. This exact thought was what led to the development of this paper. In other words, applying an attribute grammar to a context-aware system which recognises physical activities and gives suggestions according to the context. The system acts both before and during the physical activity. For example, it can suggest the user to go for a run outside if the weather presents good conditions and during the physical activity it suggests a change on the activity if something is wrong. By using this kind

of grammars in such application it is possible to get a more well-defined system in what concerns the validation of a physical activity. A physical activity is validated only if all of its sub activities are well executed.

The purpose of this work is to apply an attribute grammar to an human activity recognition platform when it comes to validate a physical activity performed by the user. The use of an attribute grammar allows to define semantic rules which should be satisfied to validate the activity. It defines what can be accepted or not. At first the Abstract Syntax Tree (AST) should be filled accordingly to the analysis of raw data. Then the system should evaluate the AST in order to validate the physical activity. The physical activity is composed of several activities. If at least one of the activities which compose the physical activity is not validated after the analysis of the grammar, a warning should be given to the user saying that the activity is not being properly executed.

The document is structured as follows. Section 2 presents concepts related to context, context-aware computing and attribute grammars. In Section 3 it is described the proposal that uses an attribute grammar for the recognition of human activity. After that, Section 4 presents the CAPAS architecture for context-aware systems, where the grammar will be applied. Following, Section 5 presents the discussion where some case studies are presented and explained. Fi-

nally, final considerations and future directions are presented in Section 6.

## 2 Background

### 2.1 Physical Activity monitorization

The current development in technology allows to detect and recognise physical activities in a really easy way. Nowadays, it is possible to recognise activities such as running, lifting weights, push-ups, cycling and many others using different sensors like accelerometers, gyroscopes and heart rate (Lara and Labrador, 2013). With that in mind it is possible to monitor physical activity in order to prevent injuries, by analysing the patterns of an user on a certain activity for example.

Currently there are some works which focus on activity and physical activity monitoring. For instance, the work (Mukhopadhyay, 2015) reviews many of the used sensors in order to evaluate whether they are efficient or not. Besides that it describes the most used physiological parameters for this purpose. For example body temperature, heart rate and blood pressure. Also, it addresses some challenges when it comes to development of an application for activity monitoring. For instance, the fact that wearables need to be light in order to be comfortable when the user uses it. Energy is something that needs to be considered as well, the device's battery should last a reasonable amount of time. This work is quite helpful since it reviews a lot of sensors and helps when it comes to choose the proper sensors. Besides that, it mentions important factors which should be taken into account when developing this application as the mentioned problems, for instance.

Another interesting work is (Capecci et al., 2016), where it was presented a smartphone based architecture with the goal to monitor patients that suffer from the disease Parkinson. That application detects the freezing of gait (FOG). Which causes a paralysis on the patient's muscles where the risk of falling highly increases. So the main goal is to detect it and warn someone whenever it happens. Even though it monitors the activity of the user as in the presented work. Its goal is not to give suggestions.

There is another related work (Angelo Costa and Julian, 2018) which monitors elderly and gives them physical activities suggestions according to their performance. It first starts by evaluating the execution of the activity. If the user performed it correctly it goes into a new activity, if he/she didn't perform it correctly the system keeps in the same activity until

it is correctly executed. This is the most similar work to the one on this paper so many things are common between both works. However, the difference is that it uses a different approach when it comes to detect the activity. Instead of using wearables it uses cameras. Video-based activity recognition systems are widely studied in the literature (Xia and Aggarwal, 2013; Ming et al., 2012; Jalal et al., 2015). However, this technique is not always the most used because it generates some distrust in the protection of the user's privacy. This is the reason why in this work, those recognition systems that use cameras to identify specific patterns are not taken into account.

What differentiates this work from previous studies is that with the attribute grammar semantic rules can be added to the model. For instance, the work (He and Jin, 2008) predicts activities with the same classification algorithm, which is support vector machine. However, a further validation of the workout will be harder. The semantic rules added by the attribute grammar, such as, a workout is composed of several activities turns the process of validation easier. For instance, in order to validate a workout, every single activity that composes that workout needs to be well executed. Therefore, just by analysing the abstract syntax tree it is possible to check if the exercises are well executed or not.

### 2.2 Context Awareness

Context awareness uses data from environments to adapt applications and services according to the changes detected there. The main goal of this process is to improve human assistance. Thus, this section aims to describe the concepts of context and context-aware computing that was adopted for the development of the research.

#### 2.2.1 Context.

As pointed in (Abowd et al., 1999), there are several definitions about the term context in the literature. Some consider it as the environment or situation, others consider it to be the user's environment while others consider it as the environment of the application. According to (Alegre et al., 2016), context is divided in two categories which are operational and conceptual. The former helps to understand issues and challenges that comes with each data acquisition technique. The latter helps to better understand the connection between contexts.

Another way to look at context is by the perspectives of active context. Which can be described as discovered context and passive context where the application presents the context to the user instantly or it

can also store it for the user to retrieve later as defined in (George Wamamu Musumba, 2016).

However, according to (Abowd et al., 1999) context-aware applications look at the *who's*, *where's*, *when's* and *what's* of entities and according to that information it is possible to know why a situation is occurring. In spite of being important variables of context, they are not enough and there are certain types of context more important than others such as **location**, **identity**, **activity** and **time**. With the former ones (who, where, when and what) the application can only know information about location and identity. However, in order to characterise a situation, activity and time information are needed. These context types allow to answer the questions of *who*, *what*, *when*, and *where*. Besides that, it also acts as indices into other sources of contextual information.

For the development of this paper it was considered the definition of (Abowd et al., 1999) which says that context is any information that can be used to describe the current situation of an entity. Where an entity can be a person, place, object or anything that can be relevant to the interaction between an user and an interface. As mentioned in (Abowd et al., 1999), context consists only of implicit information.

### 2.2.2 Context-aware computing.

Context awareness was first introduced by (Schilit and Theimer, 1994), where it was defined as "the ability of a mobile user's applications to discover and react to changes in the environment they are situated in" (Schilit and Theimer, 1994). It refers to the identification of the current state of entities and how they influence the system. However to better comprehend context awareness, it is needed a better understanding on what context is. So this section discusses a definition of context and some other aspects about context awareness.

According to (George Wamamu Musumba, 2016), context-aware computing is a mobile computing paradigm where applications discover user's contextual information such as location, time of the day and so on. Then, it takes advantage of that information in order to perform a certain activity. The work (George Wamamu Musumba, 2016) fits context in two categories the active and the passive context. Those categories lead context-aware computing to be defined in two perspectives, active context awareness which automatically adapts to discovered context by changing the application's behaviour and passive context awareness which presents the new or updated context to an user or stores it in order to use it later.

Considering that, a context-aware application needs to use the context it is inserted in, in order to

provide information or automatically execute a service, e.g., playing a song when the user arrives at a certain place or as presented in the case study (Pedro Rangel Henriques, 2018), a message is displayed on his smart phone, for instance when it is time to take medicine. According to (Schilit and Theimer, 1994) a context-aware application adapts itself to the context. Another interesting and more general definition given at (Abowd et al., 1999) says that a context-aware application uses context to not only adapt its behaviour according to it but it also uses context to display relevant information. For instance an application that only shows the temperature of a room, it is not modifying its own behaviour but it still is a context-aware application while the definition given by Schilit and Teimer (Schilit and Theimer, 1994) gives the idea that only reactive applications, like an application that turns on the air conditioner when the room is too hot, would be a context-aware application.

This application is able to give suggestions on the go due to the context-aware computing paradigm. It analyses the context the user is inserted in, fills the attribute grammar and gives the user a suggestion.

However, there are problems with context-aware applications. As pointed in (Yujie and Licai, 2010), *Evaluation* is one of those problems. The process of evaluation on a context-aware system fail to take into account many context factors. The use of attribute grammars can reduce this problem. They allow to define a set of rules in order for the context to be validated.

## 2.3 Attribute Grammar

According to (Karol, 2006), the semantics of a context-free grammar is defined by an attribute grammar. It was previously used by compiler developers in order to define the static semantics of a programming language.

Nowadays, the components used in the semantics analysis phase by compiler-generators is generated automatically by attribute grammars which take it from the user's specification. According to (Karol, 2006), usually in the process of compilation there are three phases. The first one refers to the lexical part, which converts a stream of characters into a stream of terminal symbols. The second part is the syntactical phase, which converts those terminal symbols into an attributed syntax tree. The last phase is the semantic analysis which receives an attributed syntax tree as input in order to be examined and the obtained output is the evaluated attributed syntax tree.

Still according to (Karol, 2006), an attribute grammar is an extension of the notion of context-free

grammars through two different types of attributes. The inherited attributes and synthesised attributes. The former are used in a top-down approach where it specifies the flow of information from a node while the latter characterise information in a bottom-up approach. The proposed attribute grammar in this work is composed of both inherited and synthesised attributes.

An attribute grammar is composed of non-terminal symbols, terminal symbols and productions. Productions are what define the rules of the grammar. It defines what can be accepted or not by the attribute grammar. Semantic rules express the relations between attribute values of different nodes (Karol, 2006). Another important part of an attribute grammar is the testing for cycles since it is impossible to evaluate attributes which depend on themselves. The semantic rules need to be well defined. They need to lead to definitions of all attributes at all nodes of all derivation trees (Knuth, 1968). An attribute grammar does not contain a cycle if there is no attribute A which depends on B if B depends on A.

### 3 Attribute grammar proposal for physical activity monitoring

This paper intends to present a formal validation of data related to human activities recognition. One of its goals is to present a solution for the process of validation of a physical activity performed by an user. It intends to recognise the performed activity and give instant feedback to the user whether it is being well executed or not. The data related to the activities is validated through an Attribute Grammar. Therefore the application will have three phases, the analysis of the raw data, then the grammar should verify the values accordingly to the previously analysed data and then the validation of the data present in the attribute grammar. Following it is presented the productions of the attribute grammar.

- p1:**  $physicalActivity \rightarrow id, type, activities, location, begin, duration$
- p2:**  $activities \rightarrow activity+$
- p3:**  $name \rightarrow name1, begin$
- p4:**  $activity \rightarrow name, duration, begin, execution$
- p5:**  $duration \rightarrow CT, begin$
- p6:**  $execution \rightarrow TRUE|reason$

The grammar considers that each physical activity is composed of an identification code (*id*) which identifies the physical activity, a *type* which can be for example upper if the physical activity has a focus on the upper part of the body. The *location* where it

is being performed such as a park or a gymnasium, its *beginning*, which is equivalent to the beginning of the first activity, its *duration* and a set of *activities* [p1]. Regarding the *duration* of the whole physical activity, it results from the sum of each sub activity's duration. Besides that, it is a synthesised attribute since it depends on the value of attribute duration of every single activity.

First of all the *name* of the activity results from the combination of the name of the activity and its beginning [p3]. This attribute grammar considers that a physical activity should be composed of a set of *activities* since that it is the natural evolution of a physical activity. Usually a physical activity is composed of two or more *activities* [p2]. While an activity is composed of its *name*, *duration*, *beginning* and *execution* [p4].

The duration of a sub *activity* results from the current time and the *beginning* of it [p5]. The value of the symbol *execution* can vary accordingly to the user's execution of certain activity [p6]. If the user executes the activity correctly, it should hold the value TRUE. If not it should have the *reason* why it is not correctly executed. A user performs an activity correctly when the required parameters are met which were defined during the implementation of the application. For instance during a run, if the heart rate bpm is over 85% it can be considered the wrong way, therefore this should have the reason why it is a failed execution.

The production *execution* in the attribute grammar is particularly useful on applications which evaluate the correct execution of a physical exercise. If a user is not performing it correctly the system should not validate it and then it should recommend a change in the way the user is executing it. Apart from that, it can also be possible to infer some health problem with the user or even prevent it since a wrong movement can lead into a bad injury and being able to analyse it and correct it can prevent them. The system can detect a health problem by analysing the pattern throughout the time. For instance, if the user has been performing an activity correctly and there is one day in which the system detects some anomaly it might be possible that something is wrong.

The validation of the attribute grammar should be done following some requirements. The first one is that every activity present on the physical activity should have the value of the attribute execution set to TRUE. Meaning that it was correctly executed by the user. Then it analyses the rest of the data such as location, duration and all the other parameters. If those values are between a predefined range the attribute grammar should be validated.

## 4 Context-aware system Architecture for Physical ActivitieS (CAPAS)

Following, in figure 1 it is possible to see the CAPAS architecture for context-aware systems that aim to monitor physical activities.

First, data is gathered from the user through sensors. There is all sorts of data gathered from many types of sensors such as accelerometer, gyroscope, heart rate and many more which might be useful to detect the activity. Then, the data is sent to the server through appropriated communication protocols. After that, data will go through the pre-processing phase. It is where data is taken care of. For example, null values and outliers are deleted and where other anomalies are corrected.

The next phase is the feature extraction phase. It is the most important step for the classification of the activity. The chosen features allow to differentiate the activities from each other. For example, mean, standard deviation, fourier transform and many others which might be useful for a specific activity.

Following the feature extraction step, there is the prediction of the activity, where according to the extracted features a prediction will be given. It uses a supervised approach to classify the activity which means that the system learned using labeled data (Lara and Labrador, 2013). Support Vector Machine is the used algorithm since it has an high accuracy on human activity recognition and it has been widely used for this kind of tasks for the past years (Lara and Labrador, 2013). Next, the Abstract Syntax Tree (AST) is filled according to the prediction made and the data gathered from the sensors. For the given example, which is running, the system first starts to generate an identification for the activity. Then the production *activities*, as only one activity was detected, a new production is made. First it starts to fill the *name* of it according to the activity which was predicted by the model. The *duration* is calculated as the difference between the current time and the beginning of it and for last the *execution*. It is calculated according to how the exercise was performed. It is possible to know if the exercise is correctly executed or not since for every exercise which is recognised by the model, it is trained with the most common errors in each of them. The parameter *type* is filled according to the category the exercise is inserted in. There is a pre defined database with all the exercises which can be recognised by the model and its respective category.

The last step is the validation of the physical activity. The system analyses the AST and according to the defined requirements it gives the user feedback about

the activity. In order for a physical activity to be validated, every sub activity that composes the physical activity should be evaluated. For example, in a physical activity composed of two sub activities such as bicep curl and chest press, both should hold the value true in the parameter *execution*. Otherwise, it is not validated and the user is advised to change the way it is being executed.

## 5 Discussion

This section presents three case studies aiming to discuss the partial results as well as its contributions. The first case study addresses the validation of a cardio physical activity, the second one regards a physical activity which focus on the upper body and the last one is a football match. Following in Figure 2 it is presented the Abstract Syntax Tree (AST) of an attribute grammar when the system recognises the activity of *running*. The values presented in the tree are filled accordingly to data acquired from heart rate and accelerometer sensors, since the recognised activity is *running* and those sensors are usually the chosen ones for this kind of activity (Hassan et al., 2018). After an analysis is finished on the raw data gathered from them, the system fills the attributes with the values in order to validate the physical activity.

However, for the physical activity to be validated the sub activities also need to be validated. This is done by analysing the production execution. In Figure 2 it is possible to see that it holds the value TRUE since according to the gathered data from several sensors the user is executing it correctly considering the system's requirements whether it be an heart rate beat per minute threshold, a wrong movement or any other pre defined parameter. What defines the correctness of the activity is implemented directly on the system.

Then, the validation of the whole structure is done after an analysis of the values. If they all meet the pre established requirements then it is validated. The validation means that the exercise was well executed so there is no need to warn the user. After the analysis on the sub activities that compose the physical activity it should check the other parameters and see if it corresponds to the expected ones. This way it is possible to detect any problem with the user and then it should be able to warn the user saying he is not performing it correctly.

Regarding the *location* it is particularly useful since it can be helpful when the system is recognising the activity. For example, if the user is in a park it might be possible that he/she is running. The system can also be used to monitor the user while he is at the

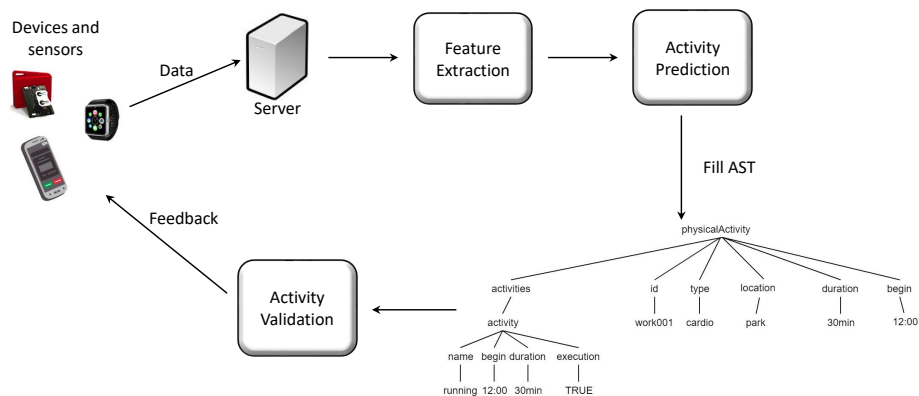


Figure 1: The system architecture CAPAS

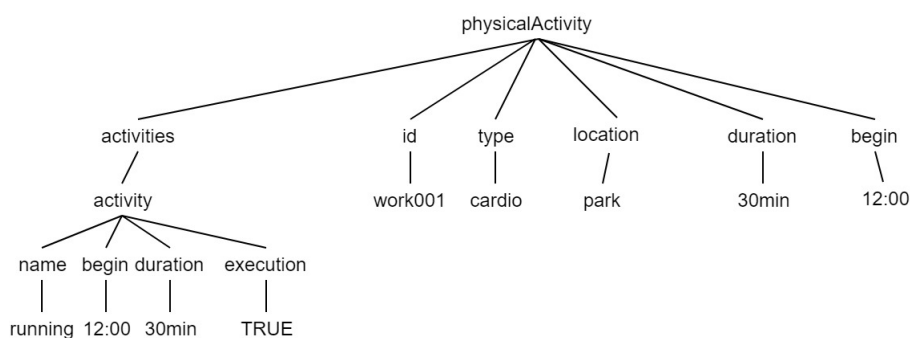


Figure 2: Attribute grammar for a cardio physical activity

gym performing a physical activity. For instance, figure 3 presents the Abstract Syntax Tree for a physical activity which focus on the upper body.

In this case, it is possible to see that this physical activity is composed of two activities, a bicep curl and a chest press. However, a physical activity can be composed by as many activities as the ones the user does.

Regarding the first activity, everything is as expected. However, the execution of the second activity holds the value *rightArmPosition*. Indicating that the user's right arm is not correctly positioned. Therefore the system will not validate the performed activity and will give a warn to the user saying that he did not perform it correctly so he/she needs to change the way. The sensors that were used to detect this activity were the gyroscope and accelerometer. The former was used to detect the position of the arms while the latter to detect the movement.

With the information that this exercise is not being correctly executed, an injury can be prevented, as already mentioned before or even have an hint that something is going on with the user speaking in terms of health. Also it can be used to monitor more general physical activities such as a football game. Figure 4

presents an AST for a football game.

Even though a football game is composed of several activities, for this example were only considered two activities, shoot and running. Since it is hard to define what a correct way is, when it comes to shoot the ball for example. More activities were not considered because the AST would be too long to be represented here.

Considering the two activities represented on the tree, they should be validated since they are considered to be executed as the right way. The validation of the activity should be done after an analysis of the tree. If the parameters are as expected it should validate it. An activity is considered validated when it is being correctly executed by the user. Besides that, the conclusion of an activity should be precise since the type of suggestions change if the user is currently in a physical activity or not. So to detect the end of an activity the system analyses the behaviour of the user and if according to some pre-defined thresholds the values are within them it considers the activity finished.

The contribution of this work to the field of study has to do with the fact that it relates the validation of activities with attribute grammars. This kind of

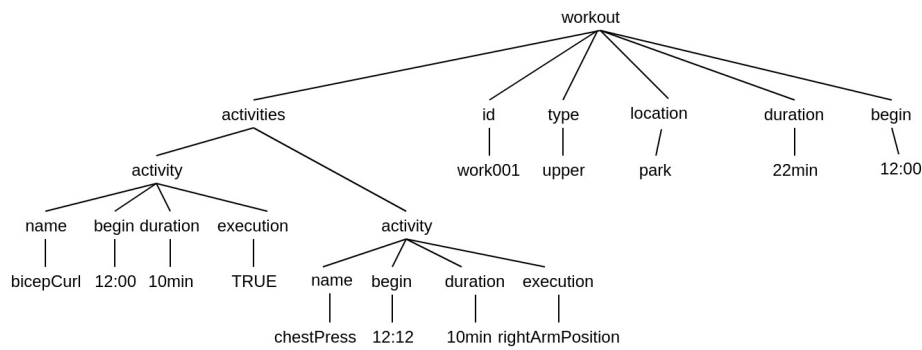


Figure 3: Attribute grammar for an upper body physical activity

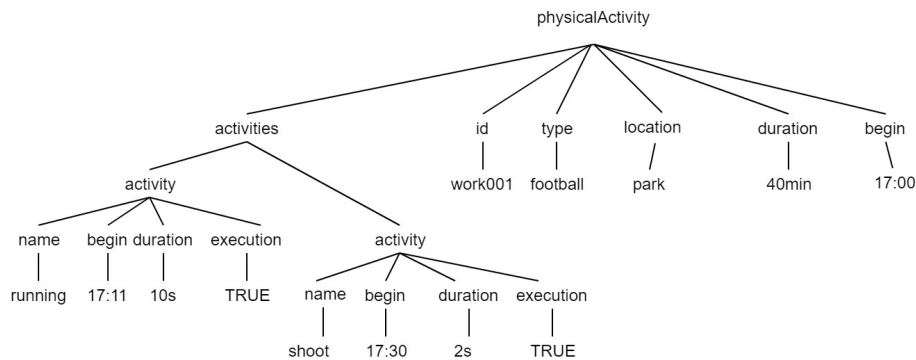


Figure 4: Attribute grammar for football recognition

application can be used in many ways not only to monitor physical activities but also to monitor elderly which are weak and need help when it comes to health care. For example if an elder falls someone should be warned about it so help can be sent. That is why this kind of applications are really useful.

However, there is a limitation regarding the addition of data from new sensors. It is not possible to process it without changing the structure of the attribute grammar. Since all the parameters should have a pre-defined source from where it's data should be collected.

## 6 Final Considerations

This paper described an attribute grammar developed to be used to validate systems for physical activity monitoring. Its productions were described in details and a context-aware system architecture, named CAPAS, for such domains was presented. Case studies are shown with a cardio and an upper body physical activities in order to let the user more familiar with the attribute grammar.

Attribute grammars applied to physical activities monitoring provides one more layer in the process of

validation which turns it into a more well defined system. They are used to validate the structure of the values of a dataset. It correlates attribute grammars and a context-aware application which has the goal to monitor and suggest physical activities according to the context the user is inserted in. Being now possible to not only monitor a physical activity but to prevent injuries as well as detect some problem the user might have.

There are still things which can be done in the future in order to improve it. For instance, expand the attribute grammar in order to be capable of accepting more than one type of physical activity in the same abstract syntax tree. It is quite usual for a person to do a specific physical activity for upper or lower body and complement it with a cardio physical activity. However, the current version of this attribute grammar treats it as two different physical activities.

Detection of health problems with the user could also be tackled. If the parameters of an exercise is different considering the user's pattern, the application could predict what is wrong with the user. Which would need to have new productions to describe the problem that happened and what it might be when talking about the human body. Future directions of research includes to analyse the efficiency of this proposal. At this point, it is not possible to do such thing.

## Acknowledgements

”This work has been supported by FCT – Fundação para a Ciência e Tecnologia within the Project Scope: UID/CEC/00319/2019.”

## REFERENCES

- Abowd, G. D., Dey, A. K., Brown, P. J., Davies, N., Smith, M., and Stegles, P. (1999). Towards a better understanding of context and context-awareness. In Gellersen, H.-W., editor, *Handheld and Ubiquitous Computing*, pages 304–307, Berlin, Heidelberg. Springer Berlin Heidelberg.
- Alegre, U., Augusto, J. C., and Clark, T. (2016). Engineering context-aware systems and applications. *J. Syst. Softw.*, 117(C):55–83.
- Angelo Costa, Ester Martinez-Martin, M. C. and Julian, V. (2018). Pharos—physical assistant robot system. *sensors*.
- Capecci, M., Pepa, L., Verdini, F., and Ceravolo, M. (2016). A smartphone-based architecture to detect and quantify freezing of gait in parkinson’s disease. *Gait and Posture*, 50.
- George Wamamu Musumba, H. O. N. (2016). Context awareness in mobile computing: A review. *Int. J. of Machine Learning and Applications*, Vol 2, No 1.
- Hassan, M. M., Uddin, M. Z., Mohamed, A., and Almogren, A. (2018). A robust human activity recognition system using smartphone sensors and deep learning. *Future Generation Computer Systems*, 81:307 – 313.
- He, Z.-Y. and Jin, L.-W. (2008). Activity recognition from acceleration data using ar model representation and svm. In *2008 international conference on machine learning and cybernetics*, volume 4, pages 2245–2250. IEEE.
- Jalal, A., Kamal, S., and Kim, D. (2015). Shape and motion features approach for activity tracking and recognition from kinect video camera. In *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*, pages 445–450. IEEE.
- Karol, S. (2006). An introduction to attribute grammars. *Department of Computer Science Technische Universität Dresden “Germany*.
- Knuth, D. E. (1968). Semantics of context-free languages. In *In Mathematical Systems Theory*, pages 127–145. California Institute of Technology.
- Lara, O. D. and Labrador, M. A. (2013). A survey on human activity recognition using wearable sensors. *IEEE Comm. Surveys Tutorials*, 15(3):1192–1209.
- Ming, Y., Ruan, Q., and Hauptmann, A. G. (2012). Activity recognition from rgb-d camera with 3d local spatio-temporal features. In *2012 IEEE International Conference on Multimedia and Expo*, pages 344–349. IEEE.
- Mukhopadhyay, S. C. (2015). Wearable sensors for human activity monitoring: A review. *IEEE Sensors Journal*, 15(3):1321–1330.
- Pedro Rangel Henriques, Paulo Novais, L. O. F. (2018). Uncertainty in context-aware systems: a case study for intelligent environments. *WorldCIST*.
- Schilit, B. N. and Theimer, M. M. (1994). Disseminating active map information to mobile hosts. *IEEE Network*, 8(5):22–32.
- Xia, L. and Aggarwal, J. (2013). Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2834–2841.
- Yujie, Z. and Licai, W. (2010). Some challenges for context-aware recommender systems. In *2010 5th Int. Conf. on Computer Science Education*, pages 362–365.