A Real-Time Entity Monitoring based on States and Scenarios

Mario José Diván

National University of La Pampa, Economy and Law School, Data Science Research Group,
La Pampa, Argentina, CP6300
mjdivan@eco.unlpam.edu.ar

and

María Laura Sánchez Reynoso

National University of La Pampa, Economy and Law School, Data Science Research Group,
La Pampa, Argentina, CP6300
mlsanchezreynoso@eco.unlpam.edu.ar

Abstract

Scenario: The current markets require online processing and analysis of data as soon as they arrive to make decisions or implement actions as soon as possible. PAbMM is a real-time processing architecture specialized in measurement projects, where the processing is guided by measurement metadata derived from a measurement framework through the project definition.

Objective: To extend the measurement framework incorporating scenarios and entity states as a way to online interpret the indicator's decision criteria according to scenarios and entity states, approaching their conditional likelihoods. **Methodology:** An extension based on entity and context states is proposed to implement scenarios and entity states. A memory structure based on the occurrence matrix is defined to approach the associated conditional likelihoods while the data are processed. A new hierarchical complimentary schema is introduced to foster the project definition interoperability considering the new concepts. An extension of the cincamipd library was carried forward to support the complementary schema. An application case is shown as a proof-of-concept. **Results:** A discrete simulation is introduced for describing the times and sizes associated with the new schema when the volume of the projects to update grow-up. The results of the discrete simulation are very promising, only 0.308 seconds were necessary for updating 1000 active projects. **Conclusions:** The simulation provides an applicability reference to analyse its convenience according to the project requirements. This allows implementing scenarios and entity states to increase the suitability between indicators and decision criteria according to the current scenario and entity state under analysis.

Keywords: Context-Aware Decision Making, Real-Time Monitoring, Measurement, Data Stream, Indicators.

1 Introduction

The General Systems Theory defines a system as a set of linked elements that interact with each other to produce an outcome and reach a common aim. Such an outcome is sent to an environment with which it interacts producing a new input known as feedback [1]. In this context, the concrete and abstract systems could be differentiated. The first ones refer to tangible systems, that is to say, those systems that are physically touchable (e.g. a person, car, etc.). The second ones are associated with non-tangible and non-physical systems (e.g. a software, equation system, etc.). Nowadays, there are two system's properties with particular importance: adaptability and dynamism. On the one hand, adaptability is understood as the system's ability to adjust its behaviour and structure in front of new contexts or changes within them. On the other hand, the dynamism is the velocity in which the system is able to adapt to the new context.

The economy could be view as an abstract system which expresses a particular behaviour depending on the country or point of view (e.g. regional or global). However, current economies could be characterized through its dynamism and adaptability, mainly thinking in the articulation between those oriented to a regional market in contrast with the

oriented to the global market [1]. The level of volatility, variations, and projections of each economy depends on a set of factors that need to be quantified to study, understand, model and forecast its behaviour. In this sense, the measurement process constitutes a key asset because it allows to establish a systematic approach to quantify an entity under monitoring, keeping measures' consistency jointly with its comparability over time [2]. From the cross-selling to the customer segmentation, the decision making is continuously tending to be an aspect related to the real-time because the people (e.g. customers and suppliers) tend to be connected [3], [4].

In this way, the measurement process is an essential part of monitoring any aspect and getting a quantitative perspective. However, this is true in the measure that it manages to be repeatable, extensible, and consistent. The repeatability implies that the same process could be applied successively in the same way always that new measures are necessities. The extensibility refers to the possibility of updating the boundaries related to the measurement itself (e.g. incorporating a new metric). The consistency is focused on how the measures are obtained, describing method, device (jointly with the precision and accuracy), among other aspects of meridian importance to warranty the measure comparability. That is to say, the measurement process must provide comparable measures due to it is essential for any kind of evolution analysis [5], [6].

A Measurement and Evaluation (M&E) framework allows to get an agreement related to the terms, concepts and the necessary relationships involved throughout a measurement process, fostering a common understanding around each monitored aspect jointly with the way in which each perspective is quantified and analysed [2].

The Processing Architecture based on Measurement Metadata (PAbMM) [7] is an Apache Storm topology that automatizes the measurement process based on a M&E framework, allowing a real-time data processing from the measures coming from heterogeneous data sources, such as Internet-of-Thing (IoT) devices.

The measurement framework associated with PAbMM uses an ontology to describe the entity to be monitored through a set of attributes. That is to say, the entity is described by a set of attributes. Analogously, the context where the entity is located is characterized by context properties, which allow a discrete approach to the environment. Both attributes and context properties are quantified using a metric. Each metric specifies the method to quantify each attribute or context property, the scale, unit, instrument (e.g. an IoT device), among other aspects. Thus, each metric is responsible to obtain a numerical value given an attribute or context property. However, the metric does not say anything about how each value needs to be interpreted, for such a reason the concept of the indicator is introduced. The role of the indicator consists of interpreting each value using decision criteria provided from previous experiences or knowledge coming from experts in the specific domain. All these concepts are required to be specified before the data arrive because they allow interpreting each value in terms of its meaning.

In this way, before a project is actively monitored by PAbMM, it needs to be defined and loaded following a specific schema. For that reason, the first stage is the project definition where the CINCAMI/PD (Context-Information Need, Concept Model, Attribute, and Indicator/Project Definition)[8] schema is used to define and describe how a given concept will be monitored. Such a project definition is interchanged between heterogeneous measurement systems to foster its interoperability.

Thus, once the project has been specified under the CINCAMI/PD schema, this is sent to PAbMM to be interpreted and loaded in memory to initialize the required resources for starting to receive measures from the IoT devices. The measures are analysed in real-time based on the informed metadata inside the project definition (e.g. the unit and scale of each measure is previously established in the definition of each metric), jointly with its interpretation that is supported by the indicators.

The original idea of the indicator in the M&E framework [2], [9] considered that it was possible to interpret a measure using given criteria independently of the current scenario or entity states. In other words, the decision criteria would be constant to interpret a measure no matter the environment or its change. However, the decision criteria are dependent on the context in which the decision should be made jointly with the current state related to the entity under monitoring. For example, to measure the heartbeat in a person who is resting is not the same that do it about a person doing an activity (e.g. walking, running, etc.). In this way, the original M&E framework was extended incorporating the possibility of modelling multiple scenarios related to the context (or environment) jointly with multiple states related to the entity under analysis to describe the convenience of given decision criteria for a certain combination of scenario and entity state.

The main contributions can be synthesized as follows: i) A new schema termed CINCAMIPD/SSA (Scenario and States Analysis) is defined as a complement of CINCAMI/PD. This makes possible to incorporate in PAbMM multiple scenarios and state definitions related to contexts and entities respectively. Also, it allows establishing the transition model related to states and scenarios with its corresponding likelihood, for supporting the multicriteria decision making. Thus, the indicator could have multiple definitions relative to the current context and entity state. It is very interesting because, in the real-time data processing, PAbMM could adapt the real-time interpretation depending on how the environment and the entity go changing; ii) A memory structure able to compute the conditional likelihood of Scenarios and States based on its frequency in PAbMM is described. The impact of the corresponding likelihoods in the compute of the mathematical expectation related to indicators is shown; iii) The CINCAMI/PD library is extended for incorporating the new CINCAMIPD/SSA schema for being able to generate and read this kind of

messages under XML (eXtensible Markup Language) or JSON (JavaScript Object Notation) data formats. Now, any heterogeneous measurement system who need multiple indicator definitions for fitting them to different contexts and entity states could do it through the updated library. The underlying idea is to foster its interoperability and extensibility; and iv) A discrete simulation analysing the consumed times and message sizes related to the library is shown.

This article is organized as follows. Section 2 outlines the related works. Section 3 describes the metadata-guided data stream processing strategy. Section 4 introduces the underlying idea related to multiple scenarios and entity states jointly with its impact the real-time data processing. Section 5 outlines the CINCAMIPD/SSA schema and the alternatives incorporated in the CINCAMIPD library. Section 6 introduces the memory structure for computing the conditional likelihood based on Scenarios and States jointly with its impact on the mathematical expectation computing. Section 7 describes the discrete simulation for the library jointly with its associated results. Section 8 introduces an application case as a proof-of-concept. Finally, some conclusions and future works are outlined.

2 Related Works

Schlosser and Boissier [10] propose a data-driven approaching for analysing aspects related to the dynamic pricing under competition in online marketplaces. Through a simulation, they analyse the way in which consumer behaviour affects the sale likelihood related to each item. They introduce the environment analysis from a multidimensional perspective, considering the multi-criteria decision making jointly with user's interactions, its profile and the sale strategy. As a difference, our strategy introduces the multi-criteria decision making based on the entity's states and their associated scenarios from the data stream viewpoint. Also, the possibility to define the entity's states and scenarios related to their contexts is based on a measurement and evaluation framework, allowing to compute conditional likelihood based on the past monitored scenarios and states.

The proposal of Wu and Xiao [11] is oriented to collect online statistics from the user experience at the moment in which they use a given application. From those statistics, the user profiles can be built, taking into consideration their interests and preferences. From the user profiles, new suggestions or learning alternatives could be carried forward. It is an interesting viewpoint, but it does not consider the effect of the user's contexts or different scenarios at the moment in which the user is using the application. For example, Using the semi-attendance system a student could attend to class from its home, or even, from the jail. In Argentina, each prisoner could study using the virtual or semi-attendance learning system, but the student's context is completely different when the contrasting between a citizen and a prisoner is carried out. For those reasons, their decisions and user profiles based on the navigation's data will be conditioned by its own context. Our proposal allows defining the states through the entity under monitoring could transit, jointly with the possibility of defining multiple scenarios which could differently affect each entity in their different states. As a similarity, both proposals employee the real-time data processing for updating its statistics and in a derived way, the associated likelihoods with each online aspect under monitoring.

The online recommender systems are an alternative for real-time recommending based on the user profile or knowledge base. In [12] an online recommendation system based on the user preferences collected from data streams is introduced. The underlying idea is to be able to adapt in real-time the recommendations based on the user preferences, but as user preferences can change along with the time, the adapts to drifts is proposed as part of the solution. The proposal uses multiple local models based on distance for analysing the belonging of each user profile to them. Because the user preference fed from the data stream could change, the belonging of a user profile is dynamic, and it can change accordingly. In [13] an online recommendation algorithm applied to online learning problem is proposed based on ensemble sampling. Kitazawa proposes an online recommending strategy based on the underlying similarity between outlier detectors and recommender systems [14]. For example, a point of change could indicate an outlier on a data stream, but at the same time, it could point to a trend in relation to a potential recommendation. As a difference, our proposal process data organized under the measurement interchange schema which allows using the metadata for detecting data quality concerns (e.g. miscalibration). Thus, it is based on the M&E project definition, the data processing is guided by metadata for analysing each entity in its current state and scenario before making a recommendation. Also, the interpretation of each measure under a given scenario and entity's state is carried out through indicators, which use the decision criteria (i.e. knowledge derived from experts) to guide the analysis before looks for recommendations. That is to say, in our strategy, the recommendation is posterior to the indicator's interpretation because the result of the indicator's interpretation guides the searching.

Wang, Yin, et.al.[15] outlines the challenges related to the online recommender systems in contrast with the traditional and offline recommender systems. In this sense, they introduce a framework termed SPMF (Stream-Centred Probabilistic Matrix Factorization Model), which is based on the Bayesian Personalized Ranking framework. Thus, the model allows dealing with situations related to new users, new items, and user's preferences may drift over time in different scenarios giving a score in terms of each context. The underlying idea is interesting because considers the change of contexts and their impact related to the user behaviour. However, this is a model that uses the data such as they arrive, without gives the due importance to the potential heterogeneity associated with the data sources and its

relative impact on the data quality. In our data processing strategy, the heterogeneity related to the data sources is incorporated both at the project definition level and the measurement interchange schema between each device and its corresponding data processor. This is a key aspect because allows properly to discriminate an outlier from a miscalibration. For example, let supposes that the data source is a thermometer continuously measuring the corporal temperature of a baby, if the thermometer gives you a value of 55 Celsius degree, it would be practically impossible in terms of human life (even when it seems to be an outlier). Such deduction is based on the metric's definition into the project definition, which specifies the device and expected variation range for the concept under monitoring (i.e. the thermometer device could produce a certain range of values with a given precision and accuracy about a person). In [16] an idea of mutual explanations is incorporated as part of the recommender system. They work around the online-dating systems and propose the idea of reciprocal explanations based on the environment related to each user. Thus, when the proposed algorithm suggests some kind of matching between two users, a reciprocal explanation is incorporated to them (i.e. a brief summary detailing the reasons which do that user 'a' match with user 'b' and vice versa). This reciprocal explanation has a good performance only when the acceptance of the recommendation has an associated cost (e.g. time, money, etc.). This proposal is very interesting because models the idea of user scenarios and try to search some kind of explanation based on the user matching itself. Our proposal is specifically oriented to the automatization of measurement processes, even though shares the idea of the impact of each scenario at the moment in which a decision must be made (e.g. establishing a matching between users).

An interesting approaching is given in [17]. The authors sustain that the recommender systems have focused on the actions related to the users in place of considering the user inaction. In this sense, they suggest that the inaction could be associated with a deliberate action from the user, or only the absence of attention. Both situations (i.e. action and inaction) could complementarily bring additional information to the recommender system, improving its accuracy and precision. In our proposal, the inaction is analysed from the point of view related to the interruption of data streams, a moment in which is necessary be able to provide approximated answers to the user based on the organizational memory (i.e. previous experiences and specific knowledge related to the entity under analysis).

Karimibiuki and Ivanov [18] have proposed a storage and query service strategy named MiniCloud, which allow keeping connected the IoT devices. In this way, each device becomes autonomous, being able to interchange data between them even when the heterogeneity is a rule in this kind of context. While the processing resources are limited to each device, the aim is put on the capacity of extensibility and interoperability related to the involved devices. Our proposal is based on heterogeneous data sources, but our data processing strategy is supported by a data stream engine, such as Apache Storm.

In [19] authors propose a recommendation framework based on user interaction. The user interaction is guided by tags, the tags are provided to users and through the different choices of them, an updated list of items is provided by the recommender system, fitting it according to the user preferences. Also, the authors propose the use of a co-occurrence analysis to refine the recommendation candidate. As a similarity, the joint occurrence is analysed as a way to detect common or typical associations in the choices. As a difference, our proposal considers the scenarios and entity states as a conditioning of the data interpretation before making a decision.

One of the challenges in recommender systems is to approach the initial user profile when they are unknown for the system, that is to say, they never have given neither data and the recommendation needs to be provided anyway. In [20], the authors propose to obtain an initial user profile using data from twitter that allows reducing the uncertainty around it. After that, they debug iteratively the profile using the user actions as inputs to improve the recommendations. As a similarity, the recommendations are based on a given user profile. However, as a difference, the proposal does not consider the context or scenario in which a given user could have made some action on twitter. Still, it is interesting how the initial profile is estimated based on a social network.

In [21] a new method for recommending, keeping a balance between accuracy and the underlying reasons (i.e. the explanation) is introduced. The idea is to incorporate better accuracy in recommendations on the online market based on automated product recommender systems. The proposal articulates the collaborative filtering based on content to deal with the recommendations. As a similarity, recommendations consider the content and its underlying meaning to increase the accuracy in recommendations. As a difference, our proposal considers the recommendations as dependant on the current scenario and entity states at the moment in which the recommendation is provided.

Table 1: A Comparative Perspective of Related Works based on their Context and Objectives

Article	Year	Context	Objective
[10]	2018	An environment analysis from a multidimensional perspective.	The support of a multi-criteria decision making based on user-interactions.
[11]	2018	The collection of online statistics from the user experience.	It provides new learning alternatives fitted to the profile.

Article	Year	Context	Objective
[12]	2018	It collects data from user preferences through the data stream reading.	It uses local models based on the distance to analyze the belonging of each user to a given profile.
[13]	2018	Online Learning.	Online recommendation based on ensemble sampling.
[14]	2017	Outlier detection and recommender systems.	An online recommendation strategy based on similarity.
[15]	2018	A context characterized by new users, new items, and aligned with user preferences.	The Stream-Centred Probabilistic Matrix Factorization Model is introduced.
[16]	2018	Online-dating systems	It describes reciprocal explanations based on the user's context.
[17]	2018	User's behavior.	An analysis of the user's actions and inactions is performed to increase the accuracy of the recommendation.
[18]	2018	Heterogeneous Devices.	It focuses on the ability to integrate and foster interoperability among heterogeneous devices through MiniCloud.
[19]	2020	Interactive Resource Recommendation.	A recommendation framework is proposed. This updates the recommendation based on provided tags by the system and the chosen tags from users. Co-occurrence analysis helps to get a better understanding of the joint behavior analysis.
[20]	2020	User profiling and Twitter.	It approaches an initial profile based on data coming from twitter.
[21]	2020	Online market. Automated product recommendation systems.	A novel method is introduced to keep a balance between the accuracy and explanation (i.e. the underlying reasons) in the online recommendations.

Table 1 presents a comparative perspective of each analysed work, highlighting the context in which each one is applied jointly with its associated aim. As it is possible to appreciate, the online recommender systems try to approach a user profile to increase the accuracy of recommendations, while they develop different strategies to decrease the risk of dealing with unknown profiles. As a general difference, the evolution of scenarios and entity states would not be explicitly indicated in them.

3 A Metadata-Guided Data Stream Processing Strategy

Measure and its associated measurement process have emerged as a natural aspect of the human to know the world surrounded them. From establishing reference patterns to lifespan analyses, the rational behaviour of the human need to quantify different aspects to learn from their context, transmit the knowledge, communicate patterns, assess structures, mitigate risks, among a lot of others applications derived from the knowledge that the measurement gives us to model our world [5]. The measurement itself is useful to represent and quantify certain aspects associated with an object, event, or abstract concept under study. That is to say, the obtained numerical value does not exist in the world, but it represents a magnitude under some scale and unit obtained through a given method using a measurement device. It sounds obvious at first glance, but it implies an agreement around the involved concepts and how a measure is got.

How a measure is got determines the possibility of the comparison, it affects the process's consistency jointly with the extensibility [2]. For example, let suppose that it is possible to measure the corporal temperature of a person in a given time. On the one hand, a person uses a mercurial thermometer using an oral method to get the corporal temperature for an entity "A". On the other hand, another person uses a mercurial thermometer using an axillary method applied to the same entity. Even when they are trying to get a corporal temperature and the values are expressed under the same range, scale and unit, both results are not comparable because the oral temperature is different from the axillary temperature with differences between 0.3°C and 0.6°C [22].

The measurement is transversal to different disciplines and industries, a reason for which there exists heterogeneity about the application environments and the involved concepts (from the measurement target up to how to interpret each value in the light of a concept). However, the measures are useful if and only if they are comparable, that is to say, let suppose that it is necessary to know whether a baby is growing up or not, the basic idea would be to measure

their weight and height in different instants comparing them with the previous ones. For reaching comparable measures, the measurement process needs to be repeatable, consistent and extensible. Consistency refers to the delimitation of the aspect being monitored, define how it would be quantified, and considering a specific method. Repeatable indicates that the process could be performed successively as many times as necessary. Extensible is associated with the possibility of updating the measurement requirements, keeping the descendant compatibility. For example, it is possible to change a measurement method but only if the new method warranties that the new measures are conceptually comparable with the previous ones (e.g. axillary versus oral temperature).

The Processing Architecture based on Measurement Metadata automatizes a measurement process using concepts, terms, and relationships specified in an underlying ontology as a way to get an agreement around the involved meanings, for example, the meaning of metric, measure, indicator, decision criteria, etc [2], [9], [23], [24]. Before incorporating the importance of scenarios and states in the decision criteria associated with indicators, this section will provide a brief introduction to PAbMM, jointly with an outline of the project definition and the measurement interchange schema.

3.1 Automatizing the Measurement Processing

The basic idea related to the data stream processing strategy is the automatization of the measurement process, consuming measures from heterogeneous data sources spread out in the field depending on the associated project. The strategy is a compound of four basic stages i) Data Collecting, ii) Data Gathering, iii) Analysis and Smoothing, and iv) Decision Making. All the stages are supported by an Organizational Memory responsible for keeping the previous experiences and knowledge from the expert associated with each measurement project.

Figure 1: Basic Behaviour of the Data Stream Processing Strategy

Figure 1 synthesizes the basic expected behaviour for the data stream strategy, describing the relationship between stages. In the beginning, the initial step consists of the loading of the measurement project, which is verified and confirmed against the organizational memory for avoiding any offset. Throughout the verification of the project is where the schema CINCAMI/PD is employed (Context-Information Need, Concept Model, Attribute, Metric and Indicator/Project Definition) to foster the interoperability among heterogeneous measurement systems. Data collecting is carried out on a mobile device implemented through a component named Measurement Adapter (MA). On the one hand, when the project does not match the project versions between MA and the organizational memory, the MA is shut down and it cannot inform any measure. On the other hand, when the project definition is the same, the MA is ready to start the data collecting. This stage is critical because the fact of share a project definition allows to know that they are analysing the same entity under analysis, the metric's specification is agreed, the devices used for collecting measures are previously known and they can be traced.

Once the data collecting has been started, it continuously provides measures collected from the associated devices which were indicated and matched using the project definition. The data gathering stage receives measures from all the measurement adapters throughout all the active measurement projects under supervising. When measures have arrived, they are replicated to the subscribers (i.e. ideal or physical systems which want to process the measures without any change), the synthesis algorithms (i.e. algorithms oriented to store a sample or meaningful changes based on the provided data in the organizational memory for future references), the statistical analysis, and the decision maker.

On the one hand, the statistical analysis performs different techniques for detecting correlations and data distributions, computing likelihoods based on frequencies, among other aspects used by the architecture when some situation needs to be analysed. In case the statistical analysis detects some abnormal situations about the general behaviour (e.g. an outlier), it immediately notifies the decision-maker for evaluating the situation. Each situation is analysed under the light of the previous experiences and knowledge stored in the Organizational Memory (OM), who is responsible for providing recommendations under the form of a scoring model. The decision-maker uses the scoring model for implementing them.

On the other hand, the decision-maker applies the incremental classifiers based on Hoeffding Decision Trees (trained initially from the previous experience in the OM) to detect typified situation modelled in the project definition (e.g. a certain set of risks). In the case of the classifier detect some typified situations will proceed in an analogous way to the statistical analysis looking for recommendations and implanting them when that be possible. That is to say, when a project is relatively recent or when the situation is not common, it is possible that there not exist previous experiences useful or knowledge useful to provide recommendations. In such cases, the scoring model tries to provide the most similar recommendations from other projects or notifying immediately to the project director for their analysis.

This is made continuously each time that a measurements' stream comes from each measurement adapter implementing the data collecting strategy, the reason for which it represents an intensive data processing environment that needs to use only the resources available for processing in-memory.

3.2 Fostering the Interoperability in the Measurement Project Definition

As it was introduced previously, the starting point to monitor any entity (physical or virtual) is related to its project definition. The project definition supposes an agreement around a set of involved concepts throughout a measurement process. That is to say, the elements that need to be used in the project definition indicate how the measurement process should be implemented. Here is where the Measurement and Evaluation (M&E) framework takes a special sense because it allows to establish the underlying concepts, terms and their relationships to get previously an agreement. For example, the understanding of the same concept for a metric, measure, an indicator as a tool for measure interpreting based on decision criteria, etc. Thus, the framework should define an underlying ontology in which fostering the agreement to sharing such concepts and relationships that next they are incorporated under the figure of a framework. For boarding this aspect, our architecture employs the C-INCAMI (Context-Information Need, Concept Model, Attribute, Metric and, Indicator) framework which defines the underlying ontology establishing the necessary relationships for implementing a measurement process [2], [9], [25].

The C-INCAMI framework has been extended from its original version incorporating complementary data [23] all of which enabled the use geographic information through GML (Geographic Markup Language)[26], add details of the data source and their associated constraints (e.g. accuracy, precision, etc.), jointly with the incorporation of audio, video, and pictures as additional information of a measure. Also, the concept of scenarios in [24] was incorporated for better support of multicriteria decision making associated with the decision-maker component.

At this point, the fact is how the project definition could be communicated and understood by the measurement systems following the directives established in the M&E framework, which is precisely the aim of the CINCAMI/PD (Project Definition) schema. The schema allows to organize the concepts, terms and their relationships under a unique XML or JSON file for fostering the interchanging of the project definitions among heterogeneous measurement systems (one of them is our data stream processing strategy mounted on Apache Storm). Once the project definition

has been communicated, each heterogeneous measurement system has the possibility of defining its own data processing strategy. Once the project definition has been communicated, each heterogeneous measurement system has the possibility of defining its own data processing strategy.

The CINCAMI/PD message contains the project definition in where each metric is defined jointly with all the involved metrics in the project (See Figure 2). That is to say, there is not a metric definition out of the CINCAMI/PD message. This is important because, with a simple self-contained file, PAbMM knows the kind of value able to arrive for a given metric. For example, the level of corporal temperature for a person could be measured through a digital thermometer, which is connected by Bluetooth with a Smart Device for informing the data collected to PAbMM. When PAbMM receives the data stream from the Smart Device, it knows the kind of expected values by using the project definition. It means that a 55°C temperature is completely abnormal in terms of the definition, being relatively easy in such cases to detect miscalibration. Figure 2.a has the Attributes tag contracted, but it is partially extended in figure 2.b for exemplifying its content.

Figure 2.a allows partially describing the definition of the metric termed “*Value of corporal temperature*”, which quantifies the “*The corporal temperature*” attribute. In the M&E project each entity under monitoring (See Outpatient in figure 2.a) answer to the information need, and it is characterized through a set of attributes (See the *Attributes* tag under the *EntityCategory* tag in figure 2.a). The quantification of the attributes will allow obtaining a quantitative point of view of their attributes, doing possible the interpretation of the measures by the indicators.

Figure 2: An Example of a CINCAMI/PD file (a) Upper Level and (b) Description of Attribute tag

Under the *metric* tag, each metric is identified (See the *IDmetric* tag in figure 2.b), jointly with the definition of the kind of scale (e.g. interval), the unit in which the measures will be expressed (e.g. Celsius degree), and the associated data source. Under the *DataSource* tag, the project defines the specific device which will provide measures for this metric. In this way, PAbMM knows the kind of expected values but also its associated data source.

Once loaded, the project definition is kept in memory for guiding the processing. Thus, in the real-time data processing, the project definition allows an immediate detection whether arrived measures are contained into the expected variation range or not without neither additional processing. Thus, an initial approach to the outlier detection (i.e. atypical values) is carried out just using the project definition.

The interpretation of each measure is the land of the indicators. That is to say, the metric knows the kind of scale, the unit, the used method for obtaining a quantitative value, the associated device which provides it, but it does not know anything about the interpretation related to the quantitative value itself. Initially, the CINCAMI/PD message incorporates a basic way to interpret the quantitative values through elementary or global indicators. The *elementary indicators* contain the decision criteria useful for interpreting the value (e.g. What 39°C represents?). The decision criteria incorporate specific knowledge from the domain's experts which is detailed under the *DecisionCriteria* tag (See figure 2.b). The global indicators allow establishing relationships between different indicators (e.g. the level of health for the outpatient).

Both global and elementary indicators are defined for the entity under monitoring (e.g. outpatient) in a given context, defined under the Context tag (See figure 2.a). In an analogous way to the entity under monitoring, the context is characterized through the context properties, which can be quantified through metrics.

Figure 3 describes an example of a measurement adapter based on Arduino One [27] used in the Bajo Giuliani's Project at Santa Rosa (La Pampa, Argentina) to monitor the level of water of a lagoon for avoiding flood in the near neighbourhoods [28]. The example allows to describe graphically the role of the project definition, data sources and measurement adapter.

Figure 3: An Example of a Measurement Adapter based on Arduino One composed of 1) Arduino One, 2) GPS, 3) Camera, 4) Real-Time Clock, 5) Wi-Fi module, 6) Ultrasonic Distance Measurer, 7) Environmental Temperature and Humidity Sensor, 8) Liquid Temperature Sensor, and 9) Soil Moisture Sensor.

Arduino One in figure 3 is connected through WiFi using the Wemos D1 Mini Pro module and keeping the time records using the RTC (Real-Time Clock) DS3231 module. In addition to the Arduino itself, there are main and complementary sensors. On the one hand, the main sensors (i.e. data sources) are focused on the quantification of defined metrics. On the other hand, the complementary sensors provide additional and optional information that could be informed jointly with the measures coming from the main sensors. It is important to highlight that each sensor provides the information such as it is, that is to say, the environmental temperature sensor (See DHT11 in Figure 3) provides numerical values but it does not know anything about the meaning of such data.

Thus, when the project definition is informed to the measurement adapter (See “(a)” in figure 3), the Arduino will match each metric from the project with its corresponding sensor (e.g. Value of the environmental temperature from the project with DHT11). Each device and metric are previously agreed upon in the project definition. When the pair sensor-metric is not matched, its data will not be provided up to the problem is solved. Next, the complementary sensors follow the same matching logic. From there, the measurement adapter is ready to provide measures for the matched pairs (metric, sensor).

Next, each sensor will provide value (e.g. See “(d)” in figure 3), while MA incorporates the corresponding tag indicating that such value is a measure associated with a metric from the project (e.g. 34.0 is associated with the metric named “Value of the environmental temperature”). This is done for each sensor linked to the MA. Once the sensor value is got, the complementary data could be incorporated (e.g. a picture and/or geographic information related to the environmental temperature to inform). Finally, data (i.e. measures) and metadata (i.e. data about the data, for example, the tags indicating the meaning for a given value) are jointly informed to the data stream engine for their processing.

In parallel, a user could send a query directly to the MA to know the last known about that specific monitoring point independently of the data stream engine. The MA will answer the queries only when the request does not compromise the available resources focused on the main sensors.

Because MA knows the project definition, it could implement certain logics oriented to optimize the communication between itself and the data processing engine, for example, load shedding techniques, dynamic switching, partial query processing, among others. In this sense, the idea is to foster communication locating this kind of device as near as is possible of users, cutting the communication distances.

3.3 Measurement's Interoperability

Each MA could gather a set of heterogeneous sensors that are designed by the project director and aligned with the measurement project's aim. The number of MA jointly with the sensors necessities for monitoring a given entity will depend on the project requirements and the area to be covered accordingly. The data stream strategy will process data coming from each MA, discriminating properly among the associated project, the measure meaning, among other aspects. For reaching it, the processing strategy uses the shared project definition for reading in each message the metadata incorporated by the MA jointly with the measures. That is to say, the tags incorporated by MA into the measurement interchange schema, allow the processor to identify the meaning of each measure (e.g. a measure from a given metric associated with a particular sensor coming from a given MA), being able to analyse and classify them on the fly. Thus, it is possible to say that data represents each measure itself, while the metadata are tags embedded in the message, which allows identifying each associated concept based on its definition.

Figure 4: A perspective of the relationship between the project definition and measurement interchange schema for the data processing.

Figure 4 outlines the underlying idea related to the Metadata-guided data processing strategy. As it is possible to see, data continuously arrive from MA through the data streams organized under the CINCAMI/MIS schema. The available processing time is limited by the available resources (e.g. memory, processor, etc.) and the new arrivals which substitute previous data. Each *MeasurementItem* tag in the CINCAMI/MIS schema bounds each measure's information. This tag indicates to PABMM the associated data source (See *dataSourceID* attribute in *measurementItem* tag in figure 4), the original data format, the footprint (i.e. a MD5 hash for verifying the integrity related to its content), the target entity (See the *idEntity* tag in figure 4), the metric responsible for obtaining the measure (See the *idMetric* tag in figure 4), and the kind of quantitative value collected for the measure (See under the *quantitative* tag, the *deterministicValue* tag in figure 4). Alternatively, the quantitative value could be a likelihood distribution in place of a deterministic value.

Thus, when PABMM read the *dataSourceID*, *idEntity* and *idMetric* tags verify the pertinence of the origin in relation to the data source, the entity under monitoring and associated metric. If in the project definition, the metric is not associated with the data source, then the measurement will be discarded, in other cases, it will continue its analysis. This is important because the project definition contains the precision and accuracy of each device related to metrics, and it allows knowing the expected range of values and the possibility of a miscalibration.

Each measure from a given data source is analysed in terms of its metric definition in real-time (See in figure 4, the lookup between the metric *dm_ctemp* in the stream in relation to the project definition in memory), which allows

identifying its unit, scale, measurement method, the associated measurement device, etc. Related to it, the indicator definition is present for interpreting each value. For example, the first value on the upper left side of figure 4 (i.e. 39.0), allows going to the indicator definition and determining that it corresponds with a risky situation for the entity under analysis, which requires a search of recommending before sending an alarm. The second value on the lower left side of figure 4 (i.e. 59.0) would correspond with a risky situation in terms of the indicator associated with the metric, however the associated data source could indicate that the value is not inside of the expected values for a corporal temperature related to a person, and for that reason a miscalibration would be detected.

This kind of contrasting and verification at the moment in which the data arrive is carried out permanently by PAbMM based on the project definition. For this reason, it is said that the data processing is guided by metadata because the own project definition allows determining whether a value is consistent or not. In this way, the rest of the posterior analysis in PAbMM uses its metadata for supporting the analysis, for example, when a descriptive statistical analysis is carried forward, it uses the metric's definition for analysing the expected values in relation to the received data series to calculate deviations, etc.

The current version of the CINCAMI/PD supposes by a simplicity that the embedded decision criteria into the indicators are independent of the current state associated with the context or entity under monitoring. This is a convenient supposition in many projects because allows keeping the parsimony in the definition. However, the states through which the context could transit (i.e. *Scenarios*) or those states in which an entity under monitoring could be exposed (i.e. *Entity's States*) could have direct incidence in the decision criteria. This could imply multiple decision criteria by each indicator depending on the pair expressed by the context's state and entity's state.

4 The Impact of the Entity and Context States

Because the interpretation of each measure is carried out through the indicators and their associated decision criteria, this aspect is a key asset at the moment in which PAbMM should make a real-time decision about the data itself and its underlying concept. The CINCAMI/PD schema introduces the indicator concept from the measurement and evaluation framework; however, a weak assumption is that each decision criteria is homogeneous along all the situations for a given metric. By example, the normal level of blood pressure [29] in a person around 30 years old could be 122/81 mmHg, but in a younger person, around 15 years old could be 117/77. Even when the magnitude could be slightly similar, the interpretation and actions could be widely different depending on the entity state and its associated context.

On the one hand, the scenarios could be expressed as a particular configuration related to the context in which the entity could be monitored. Because a context could be described through a set of context properties, the scenarios could be expressed as a particular configuration of each context property with their expected values. On the other hand, the entity is characterized through a set of attributes able to be quantified. The states represent the different stages for which an entity could go through in its lifespan. Each state could be defined by means of a specific configuration of each attribute with their expressed values.

The decision criteria capture the knowledge and previous experiences from the domain's experts, which is useful to support the decision making when a measure is analysed based on a given metric. Thus, it is important to highlight that this experience and knowledge could be dependent on particular scenarios or states. For that reason, the decision criteria could be adapted to a specific scenario, state, but also the interaction between them.

$$\forall st \in ST \wedge i \in I \Rightarrow 1 \leq |I| \leq |ST| \quad (1)$$

Where:

- **st**: It represents a particular state related to an entity under monitoring. Each state is expressed as a combination of the pair (Entity's attribute, Expected Value Range)
- **ST**: It represents a set with all the available states related to the entity under monitoring in a given project.
- **i**: It defines an indicator with its corresponding decision criteria.
- **I**: It represents a set with all the available indicators related to the entity under monitoring in a given project.

Because for each different state the decision criteria related to the Indicator could change, the Indicator could have as many definitions as states there are related to the entity under monitoring in the project. It is important to highlight that the Indicator is the only way in which a measure could be interpreted due to the incorporation of their decision criteria, for that reason the minimum quantity is 1. In other words, At least must exist one indicator for interpreting the measures. In this sense, the size of the Indicator's Set would be between 1 and the number of available states in the ST set (i.e. |ST|).

$$\forall sc \in SC \wedge ci \in CI \Rightarrow 1 \leq |CI| \leq |SC| \quad (2)$$

Where:

- **sc**: It represents a particular configuration of the environment in which the entity under monitoring is interacting. Each scenario is expressed as a combination of the pair (Context's property, Expected Value Range)
- **SC**: It represents a set with all the available context states related to the entity under monitoring in a given project.
- **ci**: It defines a contextual indicator with its corresponding decision criteria.
- **CI**: It represents a set with all the available contextual indicators related to the entity under monitoring in a given project.

However, the indicator definition could be influenced at the same time by the current entity state jointly the current environment state. For example, the situation in which an elderly man is on the beach gives a level of risk different in relation to a young woman playing in the park. Simply considering the age for defining the entity's states and the environmental temperature for the context, the decision criteria related to the level of risk for the heart could vary depending on the combination (current state, current scenario).

But even when some indicators do not share its definition (because its decision criteria are applied to different things) they could be influenced by the presence of a given scenario or entity's state. In this way, it is possible to indicate that we can have as many decision criteria as combinations of states and scenarios are possible.

$$\forall st \in ST, \forall sc \in SC, i \in I \Rightarrow i' = F(i, st, sc) \quad (3)$$

Where:

- **i'**: It represents the updated interpretation which is dependent of the indicator 'i' given a scenario and entity's state.
- **F**: It is the way in which the correspondence between i' and the original indicator, the entity's state, and scenario are established.

Equation 1 shows the maximum limit of Indicator definitions from a unidimensional viewpoint associated with the entity's states. Equation 2 describes the maximum limit of Indicator definitions from a unidimensional viewpoint associated with scenarios. However, Equation 3 allows establishing the indicators updating based on the current indicator definition (Given inside the CINCAMI/PD Message), a given state, and scenario considering the transition between them. That is to say, Each Scenario or State has a transition model which allows informing the cost, income, likelihood, and similarity for transiting from a Scenario/State to another Scenario/State. In this way, F can use the information for updating the decision criteria jointly with the provided information from the expected value range associated with scenarios and states.

The maximum limit of updating for a given indicator is established by the equation 3 as the product between the size of the sets SC, ST, and I. That is to say, the maximum number of updating for the indicators in a given project is $|I| \times |SC| \times |ST|$. For example, with 3 scenarios, 4 entity's states and 3 elementary indicators in a given project, it is possible to redefine the indicators up to $3 \times 4 \times 3 = 36$ times. Of course, each one of the 36 redefinitions could have different decision criteria because each one is in function of different triple (Indicator, State, Scenario). Thus, the total number of redefinitions is expressed as:

$$\forall st \in ST, sc \in SC, i \in I, i' \in I' \Rightarrow |I| \leq |I'| \leq |I| \cdot |ST| \cdot |SC| \quad (4)$$

Where I' represents the set of all the redefined indicators. As it is possible to appreciate in equation 4 and on the one hand, it is limited at the lower level by the current quantity of indicators originally defined in |I| (i.e. those included in the CINCAMI/PD message). On the other hand, the upper level is given by the combination associated with the triple indicated before.

It represents a key asset due to the possibility of specialization of each indicator given a scenario and state. Thus, PAbMM is able to adapt the interpretation selectively based on the triple given by indicator, scenario, and state. In case of changing of scenario and/or state, PAbMM could dynamically adapt its interpretation for a given scenario just reading the data coming from the data sources and analysing its associated metadata.

5 Interchanging the Definitions of States and Scenarios

The underlying idea related to scenarios and states is interesting, but also, we need a way in which be possible its interchanging. Also, it is necessary a complementary way for updating the original definition of indicators (jointly

with their decision criteria) once the CINCAMI/PD message had been loaded in PAbMM. For that reason, the CINCAMIPD/SSA (*States and Scenario Analysis*) schema is introduced here.

Figure 5: Initial levels of the CINCAMIPD/SSA Schema.

Figure 5.a synthesizes the highest level related to the CINCAMIPD/SSA schema. The CINCAMIPD_SSA tag groups a set of definitions related to a set of projects. It contains three attributes: i) *IDMessage*: an ID related to the message; ii) *version*: it indicates the version in which the message is expressed (e.g. 1.0); iii) *creation*: it indicates the date and time in which the message was generated. Under the main tag, there is an *SSAProjects* tag which group a set of *SSAProject* tags, which contains states and scenarios analysis. Under *SSAProject* tag there are four tags: i) *projectID*: it indicates the projects in where the indicator definition will be updated; ii) *Scenarios*: it contains all the scenarios specified for the indicated *projectID* based on its context and associated context properties (See Figure 5.b); iii) *ECStates*: it defines the available states for the indicated *projectID* based on the entity under monitoring and their associated attributes (See Figure 5.c); iv) *WeightedIndicators*: It contains each indicator redefinition based on an indicator, scenario or entity's state (See Figure 5.d).

Figure 5.b synthesizes the related structure to the *Scenario* tag. Each scenario is identified by an id, containing a descriptive name, and specifying the empirical and *theoretical likelihoods* related to its occurrence. In addition, a state transition model (See Figure 5.e) is detailed under the *StateTransitionModel* tag, which describes the possible transitions between scenarios. Finally, under the *ScenarioProperties* tag, each scenario property is characterized. The *Scenario* property contains: i) *ID*: it indicates the ID of the context property for which the particular characteristic is

defined in the scenario; ii) *name*: a descriptive name for the context property; iii) *Range*: it specifies the expected min and max values for the context property, jointly with two Boolean variables indicating if the extremes (i.e. min and max) are included in the interval.

Figure 5.c outlines the related structure to the *ECState* tag. Each entity state is identified by an *id*, containing a descriptive name, and specifying the empirical and *theoretical likelihoods* related to its occurrence. In addition, a state transition model (See Figure 5.e) is detailed under the *StateTransitionModel* tag, which describes the possible transitions between entity states. Finally, under the *ECStateProperties* tag, each state property is characterized. The state property contains: i) *ID*: it indicates the ID of the entity's attribute for which the particular characteristic is defined in the state; ii) *name*: a descriptive name for the entity's attribute; iii) *Range*: it specifies the expected min and max values for the entity's attribute, jointly with two Boolean variables indicating if the extremes (i.e. min and max) are included in the interval.

Figure 5.e shows the *State Transition Model*, tag early introduced in figures 5.b with *Scenarios* and 5.c with *entity states*. The model is composed of a set of transitions. Each transition identifies the *source ID* (i.e. the source state or scenario depending its use), the *target ID* (i.e. the target state or scenario depending its use), the *related cost* and *income* for using the transition when it is applicable, the *likelihood* related to the possibility of occurrence of it, and the *level of similarity* between the source and target states.

Figure 5.d describes the way in which the indicator redefinition is specified. Thus, under each *WeightedIndicator* tag, the decision criteria are redefined but only applicable for a given indicator, scenario and entity state (See *indicatorid*, *scenarioID*, and *ecstateID* tags in Figure 5.d). In addition, the triple composed by the indicator, scenario, and an entity's state can have in each particular configuration a given weight and parameter.

Figure 5.f establishes the *decision criteria* as a set of decision criterion which applies particularly in the combination given by indicator, scenario and entity's state. Basically, the lower and upper thresholds are established (See *lowerThreshold* and *upperThreshold* tags in Figure 5.f), indicating whether a message or alarm must be thrown when the current value is lower, between, or upper the defined interval by the thresholds (See *notifiableUnderThreshold*, *notifiableBetweenThreshold*, *notifiableAboveThreshold* tags in Figure 5.f).

In this way, with a simple CINCAMIPD/SSA message is possible to update all the definitions related to indicators for a set of projects, incorporating the scenario and entity's state definitions jointly with the state transition model. This new schema has been incorporated into the *cincamipd* library supporting concurrently the reading, writing, and message interchanging organized under CINCAMIPD and CINCAMIPD/SSA schemas

6. Approaching the Empirical Likelihood of Scenarios and Entity States

As it was introduced in figure 5.b and 5.c, the scenarios and entity states have a theoretical and empirical likelihood associated respectively.

Figure 6: Conceptualization of the Real-Time Empirical Likelihood Computing

On the one hand, the theoretical likelihood is defined by the project director as an expert in the aspect to be monitored. On the other hand, the empirical likelihood is computed in real-time considering the data arriving for its processing. However, once the empirical likelihood is known, it could be stored in the organizational memory for future use (e.g. it could provide an initial approximation to the data processor up to the current empirical likelihood be recomputed). In this way, the CINCAMIIPD/SSA schema contemplates the possibility of informing both theoretical and empirical likelihoods from and to the OM.

Figure 6 conceptually describes the real-time empirical computation for each combination of scenarios and states. When the CINCAMIIPD/SSA is informed to the data processor, a bidimensional matrix given by the size of the set of scenarios and states respectively is created (See figure 6.a). The matrix contains a set of accumulators, in which the rows represent each scenario, while the columns are the corresponding entity states. All the accumulators are initialized in zero and they represent the joint occurrence of the corresponding pair (Scenario, State) in a given project. Once the matrix has been created and initialized in memory, the data processor starts to increment the accumulators based on the measurement stream read from the measurement interchange schema coming from the measurement adapters. Each project will have its occurrence matrix that is updated independently of the rest of the projects.

While each measurement stream arrives, the measures related to each metric are read, looking up the set of related attributes to each one from the in-memory project definition (i.e. CINCAMI/PD, See figure 2). From there, the combination of attributes and their corresponding measures will determine the current entity state in terms of the definition given in CINCAMIIPD/SSA file. Analogously, the measures related to each contextual metric is read, searching for the associated context properties from the project definition. Thus, the combination of context properties and their corresponding measures will classify the current scenario based on the scenarios and states' definition (i.e. the CINCAMIIPD/SSA file).

Once the current scenario and entity state has been obtained, the accumulator in the occurrence matrix located in the intersection (Scenario, State) is unitarily incremented for representing the joint occurrence (See figure 6.c). In this way, the occurrence matrix provides complementary information as feedback of the corresponding transition models (scenarios and entity states). While the transition models keep the likelihood of transition between the same concept (i.e. between states or between scenarios), the occurrence matrix provides a different perspective because it gives the joint occurrence between a given scenario and entity state.

Thus, the occurrence matrix provides information about i) *The current empirical likelihood for a given scenario*: reading figure 6, the occurrence of "scenario2" for each entity state is 15 (i.e. $1 + 7 + 4 + 3$). The total of processed measures is 61 in this example, that implies that the empirical (i.e. based on the frequency) likelihood for the "scenario2" could be derived from the division between 15 and 61 (i.e. around 0.2459); ii) *The current empirical likelihood for a given entity state*: Analogously to the scenario, the occurrence of "state3" for each scenario is 12 (i.e. $1 + 4 + 2 + 5$). The total of measures processed is 61, implying that the empirical likelihood for the entity state "state3" is the quotient between 12 and 61 (i.e. around 0.1967); iii) *The current empirical joint occurrence for the pair (scenario, state)*: The intersection in the matrix between a given scenario and entity state (i.e. the accumulator) represents the joint occurrence of the given pair. That is to say, of the 61 processed measures, 4 times corresponded to the pair (Scenario2, State3) obtaining from its quotient the empirical joint likelihood (i.e. $4/61$, around 0.0655); iv) *The conditional likelihood for the pair (Scenario, state)*: Knowing the current entity state is "state3", what is the likelihood that the current scenario be "scenario2"? Using the Bayes's Theorem, we need the joint likelihood between "Scenario2" and "state3" (i.e. 0.0655) jointly with the likelihood related to "state3" (i.e. 0.1967). From there, only through the quotient between the joint occurrence and the likelihood of "state3", it is possible to get the conditional likelihood (i.e. $0.0655 / 0.1967 \approx 0.3329$).

The operation of updating the occurrence matrix does not represent a meaningful overhead because only it is limited to increment an accumulator in the matrix, where the access is directly performed because the current state and scenario represent the indexes of the matrix in terms of columns and rows respectively.

The occurrence matrix consumes, in general, a small memory space because its size is limited by the number of scenarios and entity states. That is to say, be "n" the size of the set related to entity states and "m" the corresponding to the set of scenarios, the dense matrix will be "mxn". It is important to highlight that the size of the corresponding sets is defined by the project director, for example, a project with 10 scenarios and 5 entity states, using an unsigned long in Java (i.e. 8 bytes) would consume 400 bytes to keep the matrix in memory, being able to represent around $2^{64}-1$ digits each accumulator.

When the matrix has processed a volume of measures near to the max's capacity (e.g. a number with $2^{64}-1$ digits), it is possible to indicate two alternatives to the data processor: i) It keeps an in-memory copy of the known state last of the occurrence matrix for reference of the data processor, reinitializing the current matrix to zero for continuing the data processing. In this sense, it is worthy to mention that this alternative duplicates the memory requirement by project (e.g. it would consume 2×400 bytes = 800 bytes), or ii) It discards the old occurrence matrix, reinitializes it to zero and indicates to the data processor that it is necessary to use the current matrix as a reference for its computations. The new indicator definitions provided through CINCAMIIPD/SSA file indicate a set of decision criteria for each combination of scenarios and entity states. However, given a value for a given indicator, it does not say anything about

how likely such value is. In other words, from the indicator data series and the likelihood from the occurrence matrix, it is possible to weigh each value using its combination of scenario and entity state.

7. Simulation Results

The original CINCAMI/PD library [8] was created for supporting the interchanging of M&E project definitions based on the common measurement framework. In this work, it is extended incorporating the ability for interchanging the new schema CINCAMIPD/SSA, which allow updating a project with new details related to the entity's states, scenarios, and the associated weighted indicators. The library is able to interchange data under XML or JSON data formats using alternatively ZIP compression. It is freely available on Github (See github.com/mjdivan/cincamipd), and it is released under the terms of the Apache License version 2.

For evaluating the times related to the library, a discrete simulation was performed on a Mac Book Pro, running macOS Mojave 10.14.4, 16GB RAM, and 500GB SSD. The simulation considered the outpatient monitoring of people, being able to walk outdoors. The simulation starts generating a CINCAMIPD/SSA message with 50 projects inside, and next, the number of projects in each message was incremented each 50 up to reach 1000 projects in one simple message.

Figure 7: A BPMN diagram synthesizing the discrete simulation process

The used procedure is synthesized in figure 7 and it could be described as follows: 1) From the context, just two context properties were considered by simplicity, the environmental temperature, and environmental humidity. From there, the scenarios Hot Temperature, Normal Temperature, Cold Temperature, High Humidity, Normal Humidity, and Low Humidity were defined; 2) The transition model for Scenarios considered that all scenarios are able to change to all scenarios except itself; 3) In relation to the entity and taking its birth date, the states were defined as young people, adult, and elderly people; 4) The transition model for Entities considered the change in a linear way without the possibility of coming back, because when a person transit from young person to adult, they do not come back to the young person stage; 5) The message was generated considering the number of projects inside given as parameter; 6) The following measures had been taken:

- The consumed time for generating the message under the XML and JSON data formats jointly with their associated sizes,
- The consumed time for compressing the message using ZIP compressor jointly with their associated sizes,
- The consumed time for decompressing the message,
- The consumed time for regenerating the object model from the XML and JSON messages.

Figure 8, on the one hand, exposes the evolution of decompression and translating times for the JSON messages on the left vertical axis (dotted lines). On the other hand, it exposes the extended and compressed sizes for the JSON messages on the right vertical axis (continuous lines). The dotted line with tiny segments represents the translating time from the JSON messages, that is to say, the consumed and necessary time for representing in a JSON data format a set de objects organized under the terms of the CINCAMIPD/SSA schema. As it is possible to appreciate in figure 8, it starts with a rate of around 0.049 (seconds/message), progressively increasing while the number of projects grow-up. However, from 650 projects atypical variations seem to be present, possibly related to the performing of the garbage collector in Java, reaching the maximum consumed time on 750 projects and not in 1000 projects such as the common sense would indicate.

The dotted line with tiny points represents the decompression time from JSON messages, in other words, the time from the reading of the compressed messages, the decompression itself up to the providing of the plain message (i.e. the text organized under the CINCAMIPD/SSA schema) for its interpretation. It starts with a rate of 0.003 (seconds/message), increasing the necessary time jointly with the number of projects. However, it experiments similar

anomalies to the previous curve, possibly related to the garbage collector because the max consumed time is reached on 950 projects and not 1000 projects such as it would be expected.

Figure 8: Message Size and Decompression Time Evolution using the JSON data format in the Processing Architecture through the *cincamipd* library

Both remaining curves associated with continuous lines on the right axis exposes the contrast between the compressed and uncompressed sizes for the same message. As a difference with the previous curves, the compression and decompression tasks did not experiment an atypical behaviour related to the garbage collector from java. In this sense, the expected behaviour of the messages' size could be approximated as relatively lineal, without apparent anomalies. It is important to highlight that the results cannot be extrapolated out of the original scopes and aims because they are circumscribed results to this simulation.

Such as it is possible to appreciate on the bottom left region of figure 8 and the one hand, 50 projects inside on a single CINCAMIPD/SSA message has a size of 1.54 MB under the JSON data format (0.02 MB compressed) and it consumes in PAbMM 0.03 seconds in the decompression jointly with 0.049 seconds in translating the JSON data format to the CINCAMIPD/SSA schema (i.e. 0.051 seconds in total). On the other hand, in the right region of figure 8, for 1000 projects the message has a size of 30.89 MB (0.36 MB compressed) and it consumes in PAbMM 0.051 seconds in the decompression jointly with 0.257 seconds in translating the JSON data format to the CINCAMIPD/SSA schema. Thus, and in this context, the possibility of redefining indicators based on their scenarios and entity's states implies just 0.308 seconds for 1000 projects. It is really promising for this kind of applications.

Table 2: Message Size and Decompression Time Evolution using XML data format in the Processing Architecture through the CINCAMIPD library

Projects	Size (MB)	Compressed Size (MB)	Decompressing Time (sec)	Translation Time (sec)
50	4.97	0.12	0.01	0.33
250	24.83	0.60	0.06	0.34
500	49.67	1.20	0.10	0.62
750	74.49	1.80	0.21	0.97
1000	99.32	2.40	0.22	1.03

The use of XML data format is supported but the sizes and times are upper than JSON data format as you can see in Table 2. For example, and in relation to the message size, while the XML data format without compression has a size

of 4.97 MB, the analogous in JSON only have 1.54 MB. In relation to the translating time, while the translating from the XML data format consumes 0.33 seconds, the analogous operation from JSON consumes 0.049 seconds (i.e. above 14 times less than XML translation). In this sense, it is evident that the preferred data format is JSON, but XML is maintained by compatibility reasons.

The times related to the translation of 50 projects from the object model underlying to CINCAMIPD/SSA to JSON and XML were 0.31 and 0.33 seconds respectively. While the compression time associated with JSON and XML were 0.71 and 2.26 seconds respectively. The average compression time by project reached in the simulation for XML was 31.64 milliseconds, while for JSON was 9.74 milliseconds. The average translation time by project reached in the simulation for XML was 1.57 milliseconds, while for JSON was 0.40 milliseconds. The use of JSON is a better alternative for this kind of application.

8. An Application Case

Following the idea introduced in the simulation, we consider an outpatient of 40 years old being monitored based on their heart rate, while the environmental temperature and humidity are collected. The first metric (i.e. the value of the heartbeat) refers directly to the entity under monitoring (i.e. the outpatient) while the last ones are associated with two characteristics of the immediate context to monitor (i.e. environmental humidity and temperature).

Thus, the heart rate is the attribute chosen to monitor from the entity under monitoring, while the context temperature and humidity represent the context properties under particular supervision for this entity.

It is possible define into the project definition, three elementary indicators for interpreting the values of each mentioned metric i) Level of the heartbeat, ii) Level of the environmental humidity, and Level of the environmental temperature.

Table 3: Partial Definition for the Indicator “Level of the heartbeat”

Indicator	Level of the heartbeat
Domain	$c \in \mathbb{N} / 0 < c < 200$
Scale	Ratio
Unit	Bpm (beat per minute)
Metric	“Number of heartbeats” using the Radial Pulse method
Decision Criteria	$< 60 \text{ bpm} \rightarrow \text{Risk}$ $[60; 100] \text{ bpm} \rightarrow \text{Low Risk}$ $(100; 120] \text{ bpm} \rightarrow \text{Normal}$ $(120; 150] \text{ bpm} \rightarrow \text{Low Risk}$ $> 150 \text{ bpm} \rightarrow \text{Risk}$

Table 3 describes partially the definition of the indicator “level of the heartbeat” for an outpatient of 40 years old. In this sense, it is important to mention that the decision criteria are general, not suitable for each scenario or entity states (i.e. in this way is how indicators are defined in the original CINCAMI/PD schema). The expected values are between 0 and 200, a magnitude that is expressed in bpm using a ratio scale. Also, it is important to pay attention to the metric to be interpreted (i.e. “number of heartbeats”) in which values are obtained through the radial pulse method. In this sense, the radial pulse was indicated, but it is not the only way to measure the heartbeat, for example, it is possible to find additional methods such as carotid pulse, pedal pulse, brachial pulse, etc. If we deliberately change the method, the concept of “Risk” given by the decision criteria could be incomparable. Thus, when updating in the indicator or metric definitions is made, it is necessary to warranty that, for example, the previous and new methods are compatibles.

Table 4: Partial Definition for the Indicator “Level of the Environmental Temperature”

Indicator	Level of the Environmental Temperature
Domain	$t \in \mathbb{R} / -10 < t < 50$
Scale	Interval
Unit	°C (Celsius Degree)
Metric	“Value of the environmental temperature” using the Heat Flow Method
Decision Criteria	$< 24 \text{ °C} \rightarrow \text{Cold}$ $[24; 30] \text{ °C} \rightarrow \text{Normal}$ $> 30 \text{ °C} \rightarrow \text{Hot}$

Table 4 partially describes the indicator “Level of the Environmental Temperature”, indicating that the expected values will be between [-10; 50] Celsius degree using an interval scale, interpreting the metric “Value of the environmental temperature” in which the values are obtained through the heat flow method [30]. The basic decision criteria were simplified for this application case, representing a value of “Cold” with environmental temperatures under 24, “Normal” with temperatures between 24 and 30, while it is indicated as “Hot” those temperatures upper than 30.

Table 5: Partial Definition for the Indicator “Level of the Environmental Humidity”

Indicator	Level of the Environmental Humidity
Domain	$h \in \mathbb{R} / 0 < h < 100$
Scale	Interval
Unit	%
Metric	“Value of the Environmental Humidity” using the Chilled Mirror method
Decision Criteria	< 50 % → Low [50; 60] % → Normal > 60 % → High

Table 5 partially defines the indicator “Level of the Environmental Humidity”, specifying that the expected values will be between 0 and 100 expressed as a percentage (i.e. the proportion of water vapor present in the air) using an interval scale and measured by the metric “value of the environmental temperature” using a chilled mirror method. “High” humidity is represented by percentages upper than 60 %, “Low” humidity is associated with values under 50%, while “Normal” humidity is related to percentages between 50 % and 60 %.

Using the attribute related to the heart rate from the outpatient, it is possible to define three states a) *Resting*: it is associated with range of heartbeats between 60 and 89 bpm, b) *Normal*: it corresponds to heartbeats between 90 and 153, and c) *In-activity*: it is related to values between 153 and 180.

Using the original interpretation of the environmental temperature and humidity, it is possible to specify three scenarios Dangerous, Caution, and Friendly such as it is indicated in Table 6.

Table 6: Scenarios Definition

<i>Temperature/Humidity</i>	High	Normal	Low
Hot	Dangerous	Dangerous	Caution
Normal	Caution	Friendly	Friendly
Cold	Dangerous	Caution	Friendly

In this way, the determination of the current scenario will be derived from the interpretation of the elementary indicators originally defined in tables 4 and 5. For example, if a value of 80% is received for the humidity jointly with 23°C for the temperature, it will configure “high” for humidity and “Cold” for temperature, accessing to table 6 the current scenario would be “Dangerous”. It is important to highlight that the original definition only is used for entering to determine the current scenario but the final decision criteria to make a decision will be updated for each indicator depending on the current scenario and entity state, as it is possible to appreciate in table 7.

Table 7: Partial Definition for the Indicator “Level of the heartbeat” based on Scenarios and Entity States

States/Scenarios:	Friendly	Caution	Dangerous
In-Activity (153; 180]	< 120 bpm → Risk [120; 153] bpm → Low Risk (153; 180] bpm → Normal (180; 190] bpm → Low Risk > 190 bpm → Risk	< 120 bpm → Risk [120; 153] bpm → Low Risk (153; 175] bpm → Normal (175; 185] bpm → Low Risk > 185 bpm → Risk	< 130 bpm → Risk [130; 153] bpm → Low Risk (153; 170] bpm → Normal (170; 180] bpm → Low Risk > 180 bpm → Risk
Normal [90; 153]	< 80 bpm → Risk [80; 90] bpm → Low Risk (90; 153] bpm → Normal (153; 160] bpm → Low Risk > 160 bpm → Risk	< 80 bpm → Risk [80; 90] bpm → Low Risk (90; 145] bpm → Normal (145; 153] bpm → Low Risk > 153 bpm → Risk	< 80 bpm → Risk [80; 90] bpm → Low Risk (90; 135] bpm → Normal (135; 153] bpm → Low Risk > 153 bpm → Risk
Resting			

States/Scenarios:	Friendly	Caution	Dangerous
[60; 89]	< 55 bpm → Risk	< 55 bpm → Risk	< 55 bpm → Risk
	[55; 60] bpm → Low Risk	[55; 60] bpm → Low Risk	[55; 60] bpm → Low Risk
	(60; 89] bpm → Normal	(60; 80] bpm → Normal	(60; 75] bpm → Normal
	(89; 95] bpm → Low Risk	(80; 90] bpm → Low Risk	(75; 85] bpm → Low Risk
	> 95 bpm → Risk	> 90 bpm → Risk	> 85 bpm → Risk

Table 7 indicates of the rows the entity states defined, incorporating the original variations range for your reference under the respective name. The columns represent the defined scenarios, while each intersection specifies the updating of the indicator “level of the heartbeat” given the pair (Scenario, Entity State). The occurrence matrix will have the same structure that table 7 with the difference that in each intersection an accumulator is held instead of decision criteria.

Figure 9: Transition Model and Associated Likelihoods for (a) The Entity States and (b) Scenarios

Figure 9 describes the transition models for the entity states and scenarios jointly with their associated likelihoods. For example, being in the “Resting” state, the likelihood to transit from such state to itself is 0.6, which is computed based on the organizational memory (i.e. previous experience) and also, it could be updated throughout the data processing following the same reasoning that the occurrence matrix. In figure 9.a the transition messages are related to actions that the outpatient could perform (for example, wake-up, train, and rest), where depending on the source state will produce a transition to a particular target scenario. Anyway, start and shutdown messages are not proper of the entity, they represent the start-up or finishing of the monitoring activity.

Messages in figure 9.b are subtly different because each ordered pair represents a given interpretation of the original indicator decision criteria for (temperature, humidity). For example, being in the “Friendly” scenario the likelihood to transit to a dangerous scenario is 0.2, which could happen due to a hot temperature with high or normal humidity (i.e. (Hot; High) or (Hot; Normal) respectively), or even, with a cold temperature and high humidity (i.e. (Cold; High)).

As it is possible to appreciate in figure 9, the indicated likelihoods refer to the transition between scenarios or states, but they do not say anything about the joint occurrence, such as that is the aim of the occurrence matrix who mix scenarios and entity states.

Intending to describe the expected behaviour of the processing architecture, figure 10 incorporates a sequence of thirteen values for the heart rate, environmental temperature, and humidity. Based on the entity states definition introduced in table 7, over the heart rate value is indicated the first letter of the corresponding entity state (e.g. R for “Resting”). Under the temperature and humidity value, an ordered pair following the original indicator definition describes the scenario configuration, for example, (N; N) represents Normal and Normal respectively that is associated with the “friendly scenario” (See Tables 4, 5, and 6). The same reasoning is applied throughout each one of the thirteen values.

Figure 10: An Illustrative sequence of the expected behaviour of the Processing Architecture

In this example, the occurrence matrix is 3x3 given by the scenarios friendly, caution, and dangerous; while the entity states are characterized by resting, normal, and In-activity. Each time a combination of entity state and scenario is received, the accumulator is incremented (See left occurrence matrix in figure 10).

Once the values have been processed, it is worthy to appreciate at least three differences with the previous processing architecture behaviour: 1) The original interpretation always would have informed a "Low Risk" following the original definition and just an "SMS" notifying the situation would have been sent. The indicator definition based on scenarios and entity states detect early the risk situation, progressively adapting the decision criteria to each scenario and entity state; 2) When the risk is detected, now it is possible to provide the last known sequence of the indicator interpretation each one based on the scenario and state at that moment (See the bottom right region in figure 10. Ordered interpretation according with the data sequence is represented by R: Risk, LR: Low Risk, and N: Normal); 3) In addition to the alarm, empirical likelihoods (related to scenarios, states, and joint occurrence) could be provided to the specialist for the alarm's analysis.

The transition models could be fitted based on the occurrence matrix. For example, the likelihood of transiting from the "Resting" state to itself was indicated as 0.6 from the organizational memory (See figure 9), while in the processed data there is not state's change, that is to say, it was 1.0 (See figure 10) because the entity always kept resting. Analogously, the likelihood related to the scenario jumps could be updated following the defined matrix in figure 9.b. For this example, twelve scenario jumps were given as follows i) "Friendly→Caution": 1, ii) "Caution→Caution": 6, iii) "Caution→Dangerous": 1, and iv) "Dangerous→Dangerous": 4.

In this way, the possibility of incorporating scenarios and states jointly with the setup of the decision criteria to each possible combination of them is introduced. This allows to reach an interpretation adjusted to the current situation, being possible to compute the joint occurrences from the same data stream. The operation does not represent a meaningful overhead such as it was analysed in the previous section. In this example, the difference between "Low Risk" and "Risk" values could imply the difference between saving a life or not depending on the information used by the specialist for deciding the course of action.

9. Conclusions and Future Works

The incorporating of CINCAMIPD/SSA (Scenario and States Analysis) allowed extending the project definition incorporating the possibility to fit decision criteria according to a certain configuration of scenario and entity state associated with the entity under monitoring. This constituted an advance in PAbMM because previously the decision criteria were constant to interpret a measure, while it did not consider the contextual situation for a given measure. Also, the architecture now can analyse the transition model among scenarios and entity states to provide new elements in the multicriteria decision making related to certain costs, incomes, likelihoods, and similarities. It is worthy to mention that the use of the CINCAMIPD/SSA is optional, keeping the descendent compatibility in the processing strategy.

The cincamipd library was extended, incorporating the complementary model for scenarios and entity states jointly with the ability to generate messages under both XML and JSON data format (using optionally the ZIP compression). The discrete simulation provided a temporal reference for contrasting the applicability of scenarios and entity states according to the measurement project requirements. As a reference, it was required 0.308 seconds consuming 0.36 MB to decompress and load the complement to the original definition for 1000 projects using JSON data format which is encouraging. Thus, the overhead for PAbMM is not representative, because the old project definition is parallelly kept active up to the new project definition is completely loaded. During such time (i.e. 0.308 seconds), the data arriving continues being processed using the current project definition up to be replaced.

The incorporation of the occurrence matrix allowed approaching the conditional likelihood of scenarios and entity states (Individual and joint occurrence) based on its frequency at the same time in which the data is processed, avoiding an overhead, and providing an empirical point of view about the theoretical probabilities informed in the transition models.

As future work, the supervised models will be suited with the new information supplied by the scenarios, entity's states, and associated transition models. Also, decision trees trained from the organizational memory could be incorporated for replacing the use of the original indicator definition in the occurrence matrix processing.

Acknowledgements

This research is partially supported by the projects Res.CD 278/2016 and 312/18 of the Economics and Law School of the National University of La Pampa.

References

- [1]. L. Von Bertalanffy, "The History and Status of General Systems Theory," *Acad. Manag. J.*, vol. 15, no. 4, pp. 407–426, Dec. 1972. DOI: 10.5465/255139
- [2]. L. Olsina, F. Papa, and H. Molina, "How to Measure and Evaluate Web Applications in a Consistent Way," in *Web Engineering: Modelling and Implementing Web Applications*, G. Rossi, O. Pastor, D. Schwabe, and L. Olsina, Eds. London: Springer London, 2008, pp. 385–420. DOI: 10.1007/978-1-84628-923-1_13
- [3]. D. Arli, C. Bauer, and R. W. Palmatier, "Relational selling: Past, present and future," *Ind. Mark. Manag.*, 2018. DOI: 10.1016/j.indmarman.2017.07.018
- [4]. A. Griva, C. Bardaki, K. Pramataris, and D. Papakiriakopoulos, "Retail business analytics: Customer visit segmentation using market basket data," *Expert Syst. Appl.*, 2018. DOI: 10.1016/j.eswa.2018.01.029
- [5]. R. OMNÈS, "Measurement Theory," in *The Interpretation of Quantum Mechanics*, 2018. DOI: 10.2307/j.ctv346qpb.13
- [6]. A. Maul, L. Mari, D. Torres Iribarra, and M. Wilson, "The quality of measurement results in terms of the structural features of the measurement process," *Measurement*, vol. 116, pp. 611–620, Feb. 2018. DOI: 10.1016/j.measurement.2017.08.046
- [7]. M. J. Diván, "Processing architecture based on measurement metadata," in *2016 5th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, 2016, pp. 6–15. DOI: 10.1109/ICRITO.2016.7784912
- [8]. M. Diván, M.; Sánchez Reynoso, "Fostering the Interoperability of the Measurement and Evaluation Project Definitions in PAbMM," in *2018 7th International Conference on Reliability, Infocom Technologies and Optimization: Trends and Future Directions, ICRITO 2018*, 2018, pp. 228–234. DOI: 10.1109/ICRITO.2018.8748766
- [9]. H. Molina and L. Olsina, "Towards the Support of Contextual Information to a Measurement and Evaluation Framework," in *6th International Conference on the Quality of Information and Communications Technology (QUATIC 2007)*, 2007, pp. 154–166. DOI: 10.1109/QUATIC.2007.21

- [10]. R. Schlosser and M. Boissier, "Dynamic Pricing Under Competition on Online Marketplaces: A Data-Driven Approach," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018, pp. 705–714. DOI: 10.1145/3219819.3219833
- [11]. B. Wu and J. Xiao, "Mining Online Learner Profile Through Learning Behavior Analysis," in *Proceedings of the 10th International Conference on Education Technology and Computers*, 2018, pp. 112–118. DOI: 10.1145/3290511.3290560
- [12]. M. Al-Ghossein, T. Abdessalem, and A. Barré, "Dynamic Local Models for Online Recommendation," in *Companion Proceedings of the The Web Conference 2018*, 2018, pp. 1419–1423. DOI: 10.1145/3184558.3191586
- [13]. X. Lu, Z. Wen, and B. Kveton, "Efficient Online Recommendation via Low-rank Ensemble Sampling," in *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018, pp. 460–464. DOI: 10.1145/3240323.3240408
- [14]. T. Kitazawa, "Sketching Dynamic User-Item Interactions for Online Item Recommendation," in *Proceedings of the 2017 Conference on Conference Human Information Interaction and Retrieval*, 2017, pp. 357–360. DOI: 10.1145/3020165.3022152
- [15]. W. Wang, H. Yin, Z. Huang, Q. Wang, X. Du, and Q. V. H. Nguyen, "Streaming Ranking Based Recommender Systems," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 525–534. DOI: 10.1145/3209978.3210016
- [16]. A. Kleinerman, A. Rosenfeld, and S. Kraus, "Providing Explanations for Recommendations in Reciprocal Environments," in *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018, pp. 22–30. DOI: 10.1145/3240323.3240362
- [17]. Q. Zhao, M. C. Willemsen, G. Adomavicius, F. M. Harper, and J. A. Konstan, "Interpreting User Inaction in Recommender Systems," in *Proceedings of the 12th ACM Conference on Recommender Systems*, 2018, pp. 40–48. DOI: 10.1145/3240323.3240366
- [18]. M. Karimibiuki and A. Ivanov, "MiniCloud: A Mini Storage and Query Service for Local Heterogeneous IoT Devices," in *Proceedings of the 8th International Conference on the Internet of Things*, 2018, pp. 21:1–21:4. DOI: 10.1145/3277593.3277622
- [19]. Q. Xie, Y. Zhu, F. Xiong, L. Li, Z. Bao, and Y. Liu, "Interactive resource recommendation with optimization by tag association and significance analysis," *Neurocomputing*, vol. 391, pp. 210–219, 2020, doi: <https://doi.org/10.1016/j.neucom.2018.12.082>.
- [20]. S. Tiwari, A. Saini, V. Paliwal, A. Singh, R. Gupta, and R. Mattoo, "Implicit Preferences Discovery for Biography Recommender System Using Twitter," *Procedia Comput. Sci.*, vol. 167, pp. 1411–1420, 2020, doi: <https://doi.org/10.1016/j.procs.2020.03.352>.
- [21]. A. Marchand and P. Marx, "Automated Product Recommendations with Preference-Based Explanations," *J. Retail.*, 2020, doi: <https://doi.org/10.1016/j.jretai.2020.01.001>.
- [22]. S. El-Sappagh and M. M. Elmogy, "Medical Case Based Reasoning Frameworks," *Int. J. Decis. Support Syst. Technol.*, vol. 8, no. 3, pp. 31–62, Jul. 2016. DOI: 10.4018/IJDSST.2016070103
- [23]. M. Diván and M. de los Ángeles Martín, "Towards a Consistent Measurement Stream Processing from Heterogeneous Data Sources," *Int. J. Electr. Comput. Eng.*, vol. 7, no. 6, pp. 3164–3175, Dec. 2017. DOI: 10.11591/ijece.v7i6.pp3164-3175
- [24]. M. J. Diván, "Extending the Data Stream Processing Strategy to Scenario Analysis," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 1.4, pp. 1–8, Sep. 2019. DOI: 10.30534/ijatcse/2019/0181.42019
- [25]. P. Becker, F. Papa, and L. Olsina, "Process conceptual base for enriching a measurement and evaluation ontology," in *CIBSE 2014: Proceedings of the 17th Ibero-American Conference Software Engineering*, 2014. https://www.researchgate.net/publication/267625472_Process_Conceptual_Base_for_Enriching_a_Measurement_and_Evaluation_Ontology [Available on January 21 of 2020]
- [26]. A. Aydinoglu, "OPEN DATA MANAGEMENT ISSUES FOR GEOGRAPHIC INFORMATION SYSTEMS," in *16th International Multidisciplinary Scientific GeoConference SGEM2016, Informatics, Geoinformatics and Remote Sensing*, 2016. DOI: 10.5593/SGEM2016/B21/S08.099
- [27]. J. Cicolani, *Beginning Robotics with Raspberry Pi and Arduino*. Berkeley, CA: Apress, 2018. DOI: 10.1007/978-1-4842-3462-4
- [28]. M. J. Diván and M. L. Sánchez Reynoso, "An Architecture for the Real-Time Data Stream Monitoring in IoT," in *Multimedia Big Data Computing for IoT Applications*, S. Tanwar, S. Tyagi, and N. Kumar, Eds. Springer Nature Singapore, 2020, pp. 59–100. DOI: 10.1007/978-981-13-8759-3_3
- [29]. A. Sherwood and R. A. Carels, "Blood Pressure," in *Encyclopedia of Stress*, 2010. DOI: 10.1016/B978-012373947-6.00057-X

- [30]. T. Olofsson, K. E. A. Ohlsson, and R. Östin, "Measurement of the environmental temperature using the sol-air thermometer," *Energy Procedia*, vol. 132, pp. 357–362, Oct. 2017.
DOI: 10.1016/j.egypro.2017.09.632