# Proposed Middleware for Sensor Networks in Cyber-Physical System Environments

Jorge R. Beingolea Garay, Alexandre M. de Oliveira and Sérgio T. Kofuji

Laboratory of Integrated Systems - LSI - Department of Electronic Systems Engineering

Polytechnic School, University of São Paulo

São Paulo, Brazil

{Jorge, amanicoba, kofuji}@pad.lsi.usp.br

*Abstract* - **This paper describes the modeling proposal of a middleware for sensor networks in Cyber-Physical (CP) environments adopting the reuse of Designs Patterns, which must favor the coupling in the underlying layers of the middleware and facilitate the communication between components in the upper layers. Our main goal is to reach the robustness and flexibility requirements as a result of the high probability of heterogeneity and critical events in Cyber-Physical System (CPS). This modeling is presented for the treatment of events by the middleware, considering functional and non-functional spatiotemporal requirements, as well as the set of requirements common to the CPS. The structure of the middleware, the MVC design pattern and the events model proposed seek to favor the abstraction, processes treatment and events that result from the CPS heterogeneous environment. The result is a lower level of complexity in implementation, simplified control of events and levels control of coupling between the components of the proposed middleware.**

***Keywords**: Middleware; CPS; WSN; Service Design Patterns; Object Design Pattern.*

## I. INTRODUCTION

A Cyber-Physical System (CPS) consists of a network of elements that interacts with physical environment and computational tools (sensoring, actuation, control and application). Most of these elements are constituted by embedded hardware devices (sensors and actuators) for monitoring and controlling of physical processes. Generally, the concept of CPS appears as an integration process of computing elements, which has become known by the physical elements and has ceased of being totally dependent. However, their autonomy and functionality will always be conditioned to other systems that extend their activities according to the monitored physical phenomenon and react in a more autonomous way, requiring, in the process, the implementation of other synchronized systems in real time.

The CPS can also be viewed as a management system capable of grouping several applications that function autonomously and include a control loop. However, for this technology to become viable for several contexts in the physical world, it is necessary to consider the limitations of computational elements that integrate this technology. For the case, such limitations become greater when it is a sensor network matter.

Recent researches on CPS showed some of the challenges on building CPS projects, and, among these challenges, we need to investigate new middleware projects for dealing with features such as heterogeneity, components scalability, and critical issues related to sensoring, control, and application. The proposals include the WebMed middleware [1], which is a Service-Oriented Architecture for CPS. On this paper, we highlight the need for more flexible and reusable middleware, taking advantage of the innovative features of web technologies. According to the heterogeneous characteristics of CPS, and their high tendency to scalability, it is easy to predict the excessive source of performance overload that will exist inside of these types of systems. In [2], a Dynamic Control Middleware (DCM) is proposed on the possibility of integration between different CPS. The proposal is limited to the control of IPV6 devices [2].

In many distributed systems, processes communicate with each other primarily through the propagation of events, which, optionally, also carries data. The propagation of events follows the basic concept of Publish and Subscribe Systems, in [7]. Events are published and middleware ensures that only processes that were associated (or, that were subscribed) for this event will receive them. In [6], the combinations and methods for treating these events correspond to an optimized real-time CORBA model. The model is optimized in its communication interface, aiming to ease the configuration and to manage aperiodic and periodic events. These are a result from underlying layers and they lead to a higher performance due to the robustness of CORBA architecture.

The diversity of applications associated with a CPS, even in a domestic environment, confirms the existence of several media, such as WLAN, ZigBee, LonWorks, and ECHONET [3][18], which difficult the interoperability protocols on different networks. To solve this problem, in [3], a service architecture for control methods is proposed. The proposal asserts that users will be able to control appliances in physical environment through intuitive operations, which react to devices changes in the physical environment. The virtual context approves the changes and performs an action tracking and service execution to a dynamic environment.

In services modeling, in the middleware structure, the unnecessary use of a component is a common occurrence when there is a weak specification of requirements on application execution flows. In [4], the proposal summary of a middleware framework made for monitoring and controlling water distribution systems in large scale is presented. The proposal highlights a middleware framework represented in a layered structure following the FOSD (Feature Oriented Software Design) model. The approach categorizes the components intrinsically or extrinsically, so the middleware can avoid unnecessary use of components in the execution flow.

Our paper describes a middleware modeling, focusing on the sensoring layer. Some of the assumptions aforementioned are discussed first and, then, some CPS characteristics and requirements that are necessary to standardize some processes are defined. This paper also describes the modeling of events abstracted by the middleware, which is capable to adapt itself to the limitations and requirements of a sensor network. The first approach is: i) this paper presents the requirements for these types of systems, ii) it discusses the layered middleware definition and iii) it discusses their implementation by reusing Designs Patterns.

This article is organized as following: In Section II, the description of specifications for cps is presented. In Section III, modeling to event's for middleware. In Section IV, overview of middleware architecture is presented. In Section V, the functional and non-functional requirements for the CPS are discussed. In Section VI, design pattern for middleware is presented. In Section VII, the structure and standardization of data are discussed and, finally, in Section VIII, conclusions and discussions are presented.

## II. DESCRIPTION OF SPECIFICATIONS FOR CPS

The description of specifications for modeling middleware begins by determining the heterogeneous set of features, which should be considered when building applications for a CPS.

- *Heterogeneity*: The diversity of sensor elements and actuators supported by a CPS demands a justified attention for the set of functional and non-functional requirements, which should comply with the wide scale of embedded hardware devices that composes the CPS application and actuation environment.
- *Hardware:* The set of devices which comes to be part of the CPS is very extensive, consisting in sensors, mobile phones, tablets, etc. However, the modeling described in this paper restricts itself to the architectures heterogeneity that composes the sensor layer, leaving control activities and actuation in the upper layers, on interface level.
- *Topology*: The topologies supported by the CPS can be various; however, the need for comprisement demands the use of topologies with high availability and reliability characteristics. The IEEE802.11s standard, in [14][15], describes variations of a mesh topology with high level of reliability, especially on the new IEEE802.16, in [16][17]. The mesh networks arise as an alternative to the improvement in wireless communication services. The network control is distributed and the network nodes can transmit and receive data while simultaneously routing through the surrounding nodes. This article restricts itself to the communication scenarios that use mesh and star networks.
- *Communication Protocols:* The diversity of devices appeals to the use of various communication protocols, which maximize the special characteristics of those hardwares. While working with wireless sensors, the set of requirements that a communication protocol should achieve begins with the mobility and low consumption of power. The middleware modeling takes in consideration the multihop and singlehop set of protocols, which predominates by its efficiency on diversity of applications in a wsn (wireless sensor network), and, as a consequence, attends to the dynamicity and comprisement characteristics of the CPS.
- *Communication standards:* The fast grow of applications for wireless communication environments has been motivating the development of various communication standards, which present their own characteristics associated to communication standards of a specific set of devices. The proposed architecture abstracts specific parameters of these communication standards, such as ZigBee, IEEE802.15.4, and others, while offer support for components association through two topologies (mesh and star).
- *Spatiotemporal Variables:* The CPS associate a large set of sensors with heterogeneous characteristics, which demand the specification of temporal requirements. These requirements must provide reliability to the application in execution and also provide the permanent monitoring of sensors, opportune processing of the information received, and the monitoring of data storage spots and events, which trigger processes in other applications.

## III. MODELING TO EVENT'S FOR MIDDLEWARE

Inside the middleware, the operations executed through components of the underlying layers are represented as events. For example, when a new *data sink*, which is associated to a specific architecture of hardware, is started along with the middleware, an event of connection is executed and managed by a "capture" method. This method operates as a set of functions that, in this case, would be the connection interface for new devices and also would identify the frame structure and invoke the service that will perform the decoding. The information is stored in a structured manner and it is accessed by the layer of services.

Each event associates a set of spatiotemporal non-functional requirements. Following this statement, we conclude that an event is defined as the combination of one or more event conditions, which are restrictions in terms of attributes, time, and localization, following the model described in [13].

$$PE_{id}\,\{t_{Eid}\,, l_{Eid}\,, V_{Eid}\,\} \qquad (1)$$

where $PE_{\text{id}}$ is the physical event identifier, $t_{\text{Eid}}$ and $l_{\text{Eid}}$ are the spatiotemporal occurrences, and $V_{\text{Eid}}$ is the physical event attribute. The observed phenomenon is represented as:

$$PO(MTid, SRid, i)\ \{t_O^o, l_O^o, v_O\} \qquad (2)$$

where $PO$ is the observed physical phenomenon; $SRid$ is the kind of sensor installed on $MTid$ (mote id); $i$ is the observed phenomenon; and $t_O^o(_{MTid,SRid,i})$, $l_O^o(_{MTid,SRid,i})$, and $v_O(_{MTid,SRid,i})$ are the occurrences of physical observation, time, place, and attributes, respectively.
The sensor event, in [13], serves as the first level for observations in the CPS model of events:

$$SE(MTid, Sid, i)\,\{t_{SE}^g, l_{SE}^g, t_{SE}^{eo}, l_{SE}^{eo}, V_{SE}, ps\} \qquad (3)$$

$MTid$ is the mote sensor that generates the event based on the identification of $Sid$ sensor event at the instance of $i$ event. Furthermore, a 6th property is used, where $t^g_{SE(_{MTid,SRid,i})}$ and $l^g_{SE(_{MTid,SRid,i})}$ are the time and place related to the occurrence of events and mote sensor, respectively; $t^{eo}_{SE(_{MTid,SRid,i})}$, $l^{eo}_{SE(_{MTid,SRid,i})}$, and $V_{SE(_{MTid,SRid,i})}$ are the estimated occurrence time to event, place and attribute, according to the mote sensor. $ps_{(MTid,SRid,i)}$ is the mote sensor degree of reliability in relation to the event sensor.

If we, for example, simplify the particular representation of an event with the 2.1 formula, the heat monitoring of the x1 server denoted as $\text{SCServer}_{\text{x1}}$ on $\text{t}_{\text{SCServer\_x1}}$ instant, located at $\text{l}_{\text{SCServer\_x1}}$ position, detected by the $\text{V}_{\text{SCServer\_Temp\_x1}}$ Temperature and $\text{V}_{\text{SCServer\_Umi\_x1}}$ Relative Humidity attributes can be expressed as following:

$$\text{SCServer}_{\text{x1}}\{\text{t}_{\text{SCServer\_x1}}, \text{l}_{\text{SCServer\_x1}}, [\text{V}_{\text{SCServer\_Tem\_x1}}, \text{V}_{\text{SCServer\_Umi\_x1}}]\} \quad (4)$$

Thus, the heat event on server x1 could be detected by the evaluation of Temperature and Relative Humidity attributes of the near sensor (x1), and could also be associated with attributes of other events defined for the system, which may be represented following the (2) and (3) formulas. The middleware events representation and forms of abstraction are described in section IV.

## IV. OVERVIEW OF MIDDLEWARE ARCHITECTURE

As represented in Figure 1, the middleware project is divided in layers and the communication happens from the superior layers to the underlying ones. In the process, each layer is responsible for offering a set of needed abstraction and integration functions between the underlying layers, taking in consideration specific requirements of the devices that are part of the application.

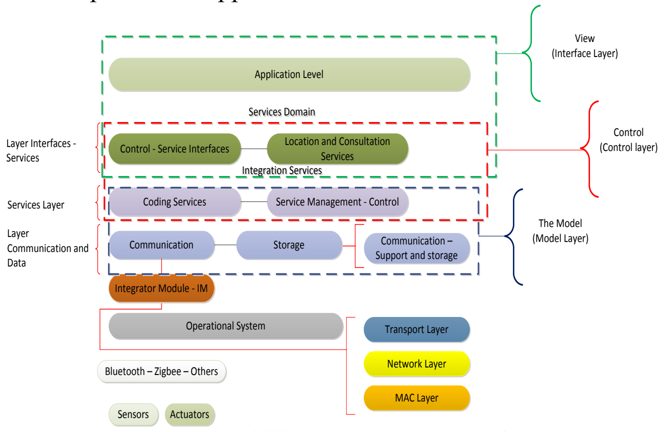

Figure 1. Middleware - Structure Layered

The basic idea for implementation of components according to a layered architecture follows a simple logic: the components defined as modular units with required and provided interfaces are organized in layers, and the components of superior layers communicate with the ones of underlying layers, but the reverse is not true. This method of implementation and communication of components optimizes the proposed middleware logic and it is oriented in the superior layers. In these layers, service requirements follow a hierarchy (the user demands a service to the inferior layers) and the results follow a vertical flow, from bottom to top.

*a) Communication and Data Layer (Communications and Support):* On this layer, the access to information must be ensured, depending on which role the user is performing during the requisition. Even so, some parameters, which are mentioned in [8], are taken in consideration in the elaboration of this module: dynamic topology, heterogeneous platforms, mixed traffic, power consumption management, and communication protocols. To achieve the quality of these parameters, it will be necessary to standardize the use of topologies and protocols for applications that should be supported by middleware, considering the diversity of CPS backgrounds.

- Integrator module (IM): The most important element for our middleware is called Integrator Module (IM). This IM acts as an element of integration between sensoring devices, actuators (*Mote Sensor T*ransmitter-MST and *Mote Sensor Actuator - MSA*), and middleware. Basically, the IM performs the capture control of network traffic, the insertion and removal of hardware modules, the abstraction of routing protocols, and the interfaces control. The topology characteristics are attended by middleware intuitively by IM, since the organization and routing parameters are transported on the protocol encapsulation at the network and link layers (Figure 1).

To ease the IM access and configuration, it is necessary to develop an API that allows the configuration of new MST values. These values are: total package size, routing package size, and data payload package size.

b) *Service Layer (Management Services):* The construction of applications, divided in communication and application modules, which resides in various computing environments, has been a great step to distributed computation. On the service integration layer, the flexibility is determined in integration with sensors network inside of a CPS, which need to receive standard technologies that offer resources to the development of new applications in return. This paper appeals to the junction and reuse of existing designs pattern and also to the use of web technologies to manage services and data both locally and remotely, considering the following features:

- *Scenario analysis:* this characteristic is indispensable because it allows, along with a requirement analysis, in [9], the determination of language, tools, and techniques that are most adequate to the abstraction of the underlying layers processes, which may satisfy objectively the application requirements to CPS, in [5].
- *Requirements classification:* On the initial requirement analysis, in [10], the total set is classified and considered obligatory, applicable, and not applicable. Following this little structured classification method, the relevance of each requirement is attributed to the computational components integration with applications through a structure based in objects and services for functional, non-functional, and spatiotemporal. This classifying method allows the determination to be made with a higher level of flexibility to relevance, coupling level and abstraction to each requirement.
- *Technologic complexity:* the selection of the most adequate integration technology is very complex due to the diversity of interfaces and programming resources that can be used, in [10]. An inadequate choice may produce serious effects in the way that services will behave at the end.
- *Standardizing*: The user receives access to the information in a consistent and easy to understand manner. Here, the use of MVC (Model View Controller) design pattern, in [11] and [12], and Facade design pattern is proposed. The reuse of such design patterns results from the need for more structured models, which should be robust and, at the same time, flexible to answer to the high CPS scalability and complexity.
- *Process Definition*: Structure definition of each stage in the process with intuit of obtaining a better classification of environment requirements for the integration middleware.

c) *Interface Layer and Domain Service Layer:* On this layer, all software parameters that automate or support the processes performed by the users on the application layer are found and grouped. Those parameters or aspects typically include each task that is part of the processes, as well as the rules and restrictions associated.

The services infrastructure is totally hybrid and should follow the basic model described in three levels:

- *Network*: The middleware architecture considers the wired means of communication, through which it gets connected with the server where the data and management modules are reposing. The wireless mean is used by devices in the acting and sensor layer.
- *Hardware*: on this infrastructure level, the heterogeneity of components and applications that make part of the middleware layer is described. Data base servers, Web and Web service, and SMS.
- *Software*: On software level, it is described the set of applications used in service domain for abstraction of process execution, resources management, and information treatment and storage.

## V. FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS FOR CPS

*a)* *Functional Requirements:* Functional requirements (Figure 2) define, in a concrete method, what the middleware should do, and they will depend on the characteristics of the application to which they are being developed, a set of minimal *requirements are presented in F*igure 2.

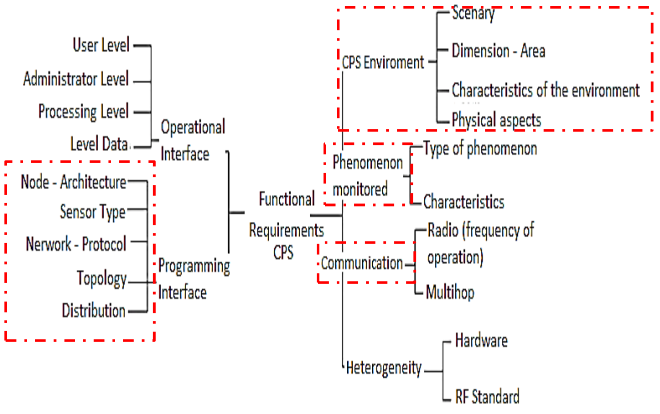

Figure 2: Functional Requirements for middleware modeling

*b)* *Non-Functional Requirements:* Before determining the set of non-functional requirements that make part of an application for CPS, it is relevant to identify the set of elements that make part of the operational interface of the hardware architecture associated to the application. As our case is a wireless sensors network, the recognition of this set of parameters allows the identification of non-functional requirements, which result of the hardware and application association, in a more clear way (Figure 3).

*c)* *Non-Functional Temporal Requirements:* The CPS associates a large set of sensors with heterogeneous characteristics. These characteristics demand a specification of temporal requirements that are able to provide reliability to the application in execution and that provide the opportune processing of information received and the monitoring of permanent sensors, storage spots of information, and events that trigger processes and other applications.

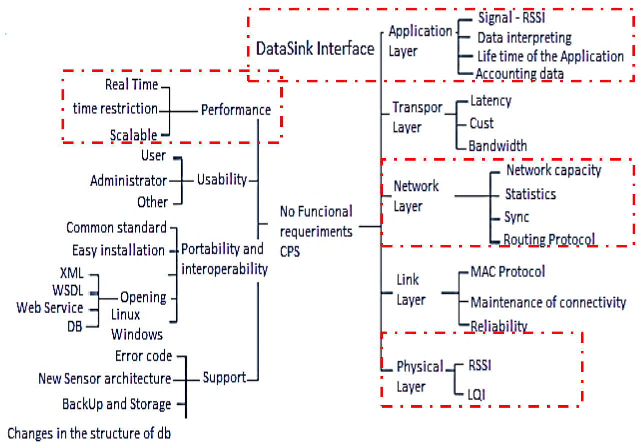

Figure 3: Non-Functional Requirements for middleware modeling

The IM connects itself to the central server and starts a connection thread with the middleware. When the connection to the server interface (data layer) is established, a second thread starts. The second thread acts as a server thread that invokes one or more worker threads.

Worker threads manage the interfaces for each data sink connected to the IM (in practice, the IM has both functions). When the first two threads start, a third one starts then and allows control and traffic classification, classifying the package flow by the communication protocol (*multihop and singlehop*) used by the sensor node (MT), as well as the type of information that it carries in the data payload, applying (6) and (7) In the classification, some non-functional requirements, non-functional spatiotemporal, and attributes extracted from the data structure entered by the data sink for IM are considered, applying (5)

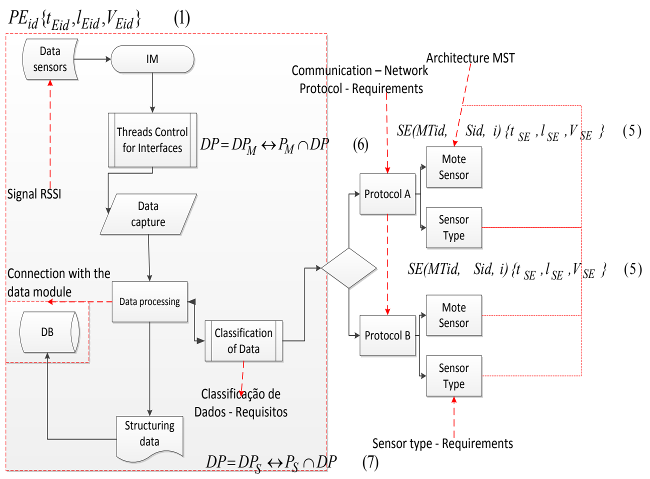
$PE_{id}\{t_{Eid}, l_{Eid}, V_{Eid}\}$ (1)
Data sensors
IM
Architecture MST
Communication – Network Protocol - Requirements
Threads Control for Interfaces
$DP = DP_M \leftrightarrow P_M \cap DP$ (6)
$SE(MTid, Sid, i)\{t_{SE}, l_{SE}, V_{SE}\}$ (5)
Mote Sensor
Protocol A
Signal RSSI
Data capture
Sensor Type
Connection with the data module
Data processing
DB
Classification of Data
Protocol B
Mote Sensor
Sensor Type
Classificação de Dados - Requisitos
Structuring data
Sensor type - Requirements
$DP = DP_S \leftrightarrow P_S \cap DP$ (7)

Figure 4: Communication Data Sink - Storage - Data Management Service - Events and Requirements

Data structure information of each data is presented as events that include non-functional spatiotemporal requirements to attribute reliability to the system. The set of events is managed by the service layer. Those sensor events and attributes are represented through a (3) simplification.

$$SE(MTid, Sid, i)\{t_{SE}, l_{SE}, V_{SE}\} \qquad (5)$$

Where SE is the sensor event, MTid corresponds to the mote architecture, Sid is the kind of sensor over the mote architecture, i is the event observed by the architecture, and $\{t_{SE}, l_{SE}, V_{SE}\}$ corresponds to the time, space, and attributes requirements, respectively.

The data classification process is performed determining common variables in the multihop protocols structure; this set of variables is represented as $P_M$, for that case, the $DP$ input data package is classified as a $DP_M$ multihop package when its structure includes a $P_M$ set of values. The same logic applies to $DP_S$ singlehop.

$$DP = DP_M \leftrightarrow P_M \cap DP \qquad (6)$$

$$DP = DP_S \leftrightarrow P_S \cap DP \qquad (7)$$

The CPS presents a high complexity and heterogeneity of hardware components and applications, which should interact with the physical environment observing, processing, and reacting to external phenomenon in real time. Following those requisitions, it is necessary to make an association with the events model (described in section II) and temporal requirements of periodic character: *i) according to the activation scheme* and *ii) according to execution time.*

- *According to the activation scheme*: These requirements could be of periodic or non-periodic activation. The periodic activation requirements are set one per T period, and exactly on each T period (Figure 5).

Inside the CPS, the wsn needs to use sampling techniques to offer a greater efficiency and, mainly, reliability of the data collected during the monitoring. On the middleware, among the temporal requirements, we have a differentiated and fixed number of samples collected per sensor. For example, considering the use of two routing protocols, singlehop and multihop, to equilibrate the efficiency and reliability of the acquired data in the physical environment, the sensors transmit the result of 5 to 10 readings in a single frame on each T. In practice, we would have 5 to 10 readings at each 100ms with a transmission cycle of up to 1000ms. This last parameter can be altered by the administrator according to the characteristics of the application. The possibility of altering the transmission cycle is interesting for some applications; when there is already a great set of samples per frame, there is no need for a high periodicity transmission.

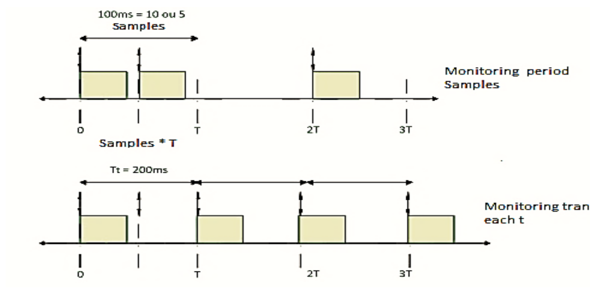

Figure 5: Representation of requirements for periodic activation.

As mentioned before, some applications in CPS, such as the control events, in [7] and [13], may demand the use of nodes to perform observations and periodic transmissions of the detected events characteristics. The degree of correlation between the consecutive mediations of the sensor node may vary according to characteristics of the physical phenomenon temporal variation, with which the CPS interacts.

In the middleware project for the CPS, it is necessary to flexible the rigor of some requirements execution, leaving the objectives of each one of them to be reached, also, in function of the best effort. Here, it is needed to consider the application of temporal requirements according to the execution time.

For that, it is possible to use some combinations of requirements, like periodicity, execution and validation time, and even the junction of temporal and functional requirements, which demands a lower level of precision, but allows the conclusion of the process. Inside the diversity of applications in CPS, the delivery and reception of data is what matters in some situations, and not the time spent in the route of reception and transmission.

- *Temporal Requirement according to execution time:* On the application of temporal requirements according to the execution time (Figure 6), it is necessary to emphasize that there are applications such as the ones of critical systems that does not admit, under any circumstances, the violation of the event implementation period.

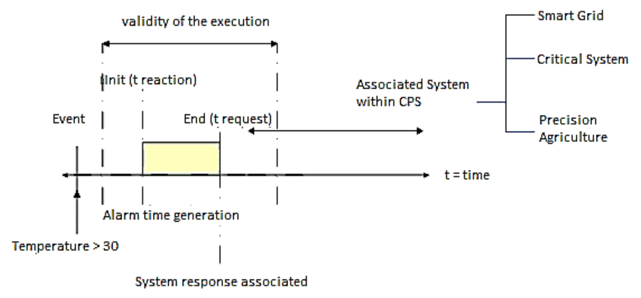

Figure 6: Representation of temporal requirements according to execution time.

The functional and non-functional requirements presented are the result of a careful study of the CPS characteristics and the wsn operating structure. The choice and determination of new requirements must pass through a specific and detailed assessment of the current architectures for sensor networks, and of spatial and physical characteristics of integration and interaction with the physical environment, which are common to all CPS.

## VI. DESIGN PATTERN FOR MIDDLEWARE

The implementation of a new or reused existing design pattern to build a middleware for CPS results in an extensive survey of new issues due to the high complexity of the application scenario that presents this new concept of systems. The construction of CPS follows a distributed logic and, as a consequence, the proposed structure for managing the diversity of components and services that exist inside of a CPS cannot be different.

The middleware initial structure must follow design patterns that allow a comprehensive description and functional requirements expression of the CPS customer and the representation of the predictable behavior of such application in execution time. The purpose of using MVC (*Model-View-Controller)* and facade is to explore the advantages of existing *design patterns* to produce a reusable middleware structure and easy maintenance for a CPS on sensor level. The MVC model adopted is organized in three layers (Figure 7), in which the intermediate layer exerts the control services and determines the application behavior.

Figure 7 shows the three-layer MVC model, which integrates the Facade Design Pattern that attributes important features to the middleware for reuse and easy maintenance. Both are models of object-oriented design that give a greater level of coupling and allow the modular modeling of each middleware components, which are associated through a service layer. Figure 7 presents the macro modeling architecture that describes the Model, the View, and the service, and also includes the DAO (Data Access Object) design pattern used to determine the access rules to data structured system.

The *design patterns* used are an excellent option to manage the middleware components in lower layers, providing a modeling that is structured and flexible enough to be reused by services architecture at the highest middleware level.

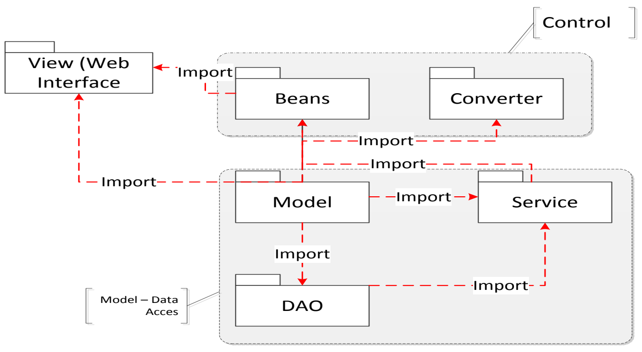

Figure 7: Specification diagram - packages abstraction

The dependency between components is defined as a result of the model that was used. The object-oriented Designs Patterns require a higher level of coupling between components, what is ideal for components that perform the abstraction and communication of data on the underlying layers, and arise from the diversity of architectures and protocols that are part of CPS. The strong dependence on middleware lower layers favors the system robustness, which is required by the high granularity of data resulted from these layers.

The coupling on upper layers is associated with service-oriented Design Pattern to web technologies, allowing the determination of some services engagement levels that are best suited for certain components, specifically those necessary to promote the reliability and availability of the system, positioning the service as a resource highly available and accessible, inclusively being required by the high willingness to critical events in the CPS.

## VII. STRUCTURE AND STANDARDIZATION OF DATA

The information is defined as structured in a relational database, however, given its heterogeneity, it is impossible to put all the information into a carefully specified database. Routing information and others that are specific to hardware are collected casually, before it has become known how they will be managed. This type of data should provide, in some cases, a variable structure. Some of the attributes contained in such data can be shared between existing entities, but some attributes may exist only in some entities. These are classified as semistructured data. The structured model and semistructured data suggest the file structure adjustment to a format that can be shared and reused by other applications. This is the case of XML data language. XML files structures are suggested for middleware, extracting information from structured database that determines common attributes of the used communication protocols and collected information inside of the CPS. When there is some set of unstructured data, these can be stored on a standard structure, so they can be analyzed later.

## VIII. CONCLUSION AND DISCUSSIONS

Although there are increasing researches on CPS, the results achieved in relation to this new concept of hybrid systems are few. The reason for this is that we still cannot count on a mature science to support the features and requirements of CPS engineering systems with a high level of reliance, and the consequences are profound. The current analysis tools are unable to cope with the high level of CPS complexity, or at least adequately predict the behavior of these systems. A clear example would be the recent events in the electrical grid, blackouts on large regions as a result of widespread failures in the system. This probably illustrates the shortcomings and limitations of current technologies, which complicates the conceptualization and creation of independent projects of engineering systems that interact with the natural and the physical world. At the same time, the demands of today's governments to develop renewable technologies, such as energy, advanced manufacturing of smart materials, vehicles powered by electricity, air transportation etc., create unprecedented opportunities to rethink many of existing systems as parts of a CPS.

The proposal of work introduced in [1] presents some lacks in relation to the integration of communications standards specified in the physical layer to support the heterogeneity of communication patterns and determine the scaling characteristic of devices in WebMed. Unlike that, the middleware architecture proposed and described in this paper intends to concentrate the devices heterogeneity result, which exists on CPS, through the middleware’s communication layer that couples the IM module. And it also intends to concentrate the management of scalability aspects that result from the adhesion of heterogeneous hardware in current implementation. For this porpoise, this paper has started describing the layers of the proposed middleware and the set of services restricted to each layer using design patterns to implementation, which is called MVC. Creating middleware structures using existing design patterns favors easy and quick system maintenance and can be extended beyond sensoring level. The middleware implementation using MVC, facade, and DAO design patterns facilitates the implementation and alteration of each layer’s functionalities, allowing the middleware to behave itself as a supervisory system on the application layer.

On CPS, the construction of each event and corresponding action happens and uses resources in a particular context. This information is fundamental to know which agents should be included in the action flow, so that the desired task is corrected in its own context, considering restrictions offered by the CPS application environment. The formalization and modeling of events, adapted from [13] and represented along with requirements according to the activation scheme and execution time, are easily abstracted from the user by design patterns without affecting the performance and transparency of the application.

The Design Patterns proposed in this paper for the middleware development follow the object-oriented and services models to balance the coupling of components and save on functional and non-functional requirements to reduce processing demands and application energy. The CORBA architectures are characterized by the variety of applications that can be created and integrated on them. However, their performance requirements, their excessive exchange of messages between components to complete a process, or their interruption on execution time let the architecture restricted to application scenarios with low mobility and greater processing resource. Therefore, CORBA architecture is not a good alternative to applications that manage hardware architectures with restricted characteristics.

## REFERENCES

[1] Hoang, d. D.; Paik, H.-Y.; Kim, C.-K. “Service-oriented middleware architectures for cyber-physical systems”. IJCSNS International Journal of Computer Science and Network Security, vol. 12, no. 1, pp. 79–87, Jan 2012.

[2] Park, S. O.; Do, T. H.; Jeong, Y.-S.; Kim, S. J. “A dynamic control middleware for cyber physical systems on an ipv6-based global network”. International Journal of Communication Systems, John Wiley and Sons, Ltd, pp. n/a–n/a, Dec 2011. doi: 10.1002/dac.1382.

[3] Lai, C.-F.; Ma, Y.-W.; Chang, S.-Y.; Chao, H.-C.; Huang, Y.-M. “Osgi-based services architecture for cyber-physical home control systems”. Computer Communications, vol. 34, n. 2, pp. 184 – 191, February 2011. http://dx.doi.org/10.1016/j.comcom.2010.03.034.

[4] Mudasser, I.; Beng, L. H. “A cyber-physical middleware framework for continuous monitoring of water distribution systems”. In: Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys ’09). New York, USA:, pp. 401–402. 2009. doi=10.1145/1644038.1644122

[5] Wood, A.; Stankovic, J.; Virone, G.; Selavo, L.; Zhimin H.; Qiuhua C; Thao D; Yafeng W; Lei F; Stoleru, R.; , "Context-aware wireless sensor networks for assisted living and residential monitoring," Network, IEEE , vol.22, no.4, pp.26-33, July-Aug. 2008. doi: 10.1109/MNET.2008.4579768. URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4579768&isnumber=4579761

[6] Zhang, Y.; Gill, C.; Lu, C. “Reconfigurable real-time middleware for distributed cyberphysical systems with aperiodic events”. Distributed Computing Systems, International Conference on, IEEE Computer Society, Los Alamitos, CA, USA, vol. 0, pp. 581–588, 2008. doi:10.1109/ICDCS.2008.96

[7] Eugster, P. T.; Felber, P. A.; Guerraoui, and R.; Kermarrec, A.-M. “The many faces of publish/subscribe”. ACM Comput. Surv., New York, USA, vol. 35, no. 2, pp. 114–131, June. 2003. doi.acm.org/10.1145/857076.857078.

[8] Xia, F.; Ma, L.; Dong, J.; Sun, Y. “Network QoS management in cyber-physical systems”. In: Proceedings International Conference on Embedded Software and Systems Symposia. Washington, DC, USA: IEEE Computer Society (ICESSSYMPOSIA ’08). pp. 302–307. July 2008. 10.1109/ICESS.Symposia.2008.84.

[9] Romer, K.; Mattern, F.; , "The design space of wireless sensor networks," Wireless Communications, IEEE , vol.11, no.6, pp. 54- 61, Dec. 2004. doi: 10.1109/MWC.2004.1368897.

[10] Cheng, B.H.C.; Atlee, J.M.; , "Research Directions in Requirements Engineering," Future of Software Engineering, 2007. FOSE '07 , vol., no., pp.285-303, 23-25 May 2007.doi: 10.1109/FOSE.2007.17.

[11] Mcheick, H.; Yan Qi; "Dependency of components in MVC distributed architecture”. Electrical and Computer Engineering (CCECE), 2011 24th Canadian Conference on , vol., no., pp.000691-000694, 8-11 May 2011 doi: 10.1109/CCECE.2011.6030542.

[12] Tao P; Lianying S; Hong B; , "Design and implementation of ATM simulation system based on MVC pattern". Educational and Information Technology (ICEIT), 2010 International Conference on, vol.1, no., pp.V1-328-V1-331, 17-19 Sept. 2010. doi: 10.1109/ICEIT.2010.5607693.

[13] Tan, Y; Vuran, M. C. and Goddard, S, “Spatio-Temporal Event Model for Cyber-Physical Systems”, In Proceedings of the 2009 29th IEEE International Conference on Distributed Computing Systems Workshops ICDCSW (2009). pp 44-50. Nov 2009. doi:10.1109/ICDCSW.2009.82

[14] Shyy, D.J.; “Military Usage Scenario and IEEE 802.11s Mesh Networking Standard”. Military Communications Conference, MILCOM 2006. IEEE , vol., no., pp.1-7, 23-25 Oct. 2006. doi: 10.1109/MILCOM.2006.302448

[15] Safonov, A.; Lyakhov, A. and Sharov, S.; “Synchronization and beaconing in IEEE 802.11s mesh networks”. Telecommunications, ICT 2008. International Conference, vol., no., pp.1-6, 16-19 June 2008. doi: 10.1109/ICTEL.2008.4652690

[16] Tae-Hwan K and In-Cheol P; “Low-Power and High-Accurate Synchronization for IEEE 802.16d Systems”. Very Large Scale Integration (VLSI) Systems, IEEE Transactions on, vol.16, no.12, pp.1620-1630, Dec. 2008. doi: 10.1109/TVLSI.2008.2001567

[17] Min-Seok Kang; Jaeshin Jang; “Performance evaluation of IEEE 802.16d ARQ algorithms with NS-2 simulator”. Communications, APCC '06. Asia-Pacific Conference on , vol., no., pp.1-5, Aug. 2006 doi: 10.1109/APCC.2006.255785

[18] Garay, Jorge R. B and Kofuji, S.T., “Architecture for sensor networks in cyber-physical system”. Communications (LATINCOM), 2010 IEEE Latin-American Conference on , vol., no., pp.1-6, Sept. 2010 doi:10.1109/LATINCOM.2010.5641126.

**Jorge R. Beingolea Garay** is B.Sc. in Computer Science (2004) - Universidad Inca Garcilaso de La Vega, M.Sc. in Electrical Engineering (2007) - University of São Paulo, and PhD. in Electrical Engineering (2012) - University of São Paulo. He is currently a Group PAD Researcher in the Integrated Systems Laboratory (LSI) at EPUSP. He has experience in Electrical Engineering, with emphasis on Wireless Sensor Networks, Wireless Communication, Computer Architectures, Wireless Networks and Pervasive Computing, Service-Based Architectures, and Cyber-Physical System.

**Sérgio T. Kofuji** is B.Sc. in Physics - University of São Paulo (1985), M.Sc. in Electric Engineering - University of São Paulo (1988), and PhD. in Electric Engineering - University of São Paulo (1995). He is currently a Group Chief Researcher in PAD of the Integrated Systems Laboratory (LSI) at EPUSP.

**Alexandre M. De Oliveira** is B.Sc. in Electrical Engineering with Computer emphasis - Catholic University of Santos (2008), M.Sc. in Electrical Engineering (2012) - University of São Paulo, and a graduate student in the University of São Paulo. He is currently Group PAD Researcher in the Integrated Systems Laboratory (LSI) at EPUSP and professor at Unimonte and Faculdade Praia Grande, with experience in Electrical Engineering, with emphasis on VLSI Design, UWB I-Radar, Timed-array propagation.