Analysis, Review and Development of a Conceptual Model, based on Class Diagrams as a Component of UML, Focused on Industrial Automation

ERIK F MÉNDEZ GARCÉS, GABRIELA M MAFLA, FRANCISCO REYES
Department of Systems
University of the Andes UNIANDES
Ave. “La Lorena”-Santo Domingo
ECUADOR
us.erikmendez@uniandes.edu.ec, gabriella_1912@hotmail.com, freyes@institutos.gob.ec

Abstract: - This document presents a detailed description of the basic structuring of the usually called "class diagrams", one of the main and most widely disseminated components of the UML (Unified Modeling Language) proposal, which seeks to deepen the contents related to computational languages applied to the industry of industrial automation, industrial systems of automatic control and the field of industrial instrumentation, linking computer science for this purpose, specifically the area of applied computing, in order to investigate alternative solutions that structure processes of nature industrial, thus strengthening the applied field of industrial computer science, emphasizing continuous production processes, such as the packaging of raw materials in the food industry, emphasizes that the current proposal is intended to present in a clear and detailed all they required for the development of a class diagram, likewise being made to include a practical connotation integrating automation main axis, this investigation.

Key-Words: - UML, Class Diagram, Automation, Control Systems, Industrial Programming, Model System.

1 Introduction

The "Unified Modeling Language" (UML: Unified Modeling Language). It has become a standard to represent, document and model structural information linked to the development of a system [1, p. 38], making emphasis in the phases of analysis and structural design of technical-industrial system architectures, this standardized language is currently a preliminary phase, for the algorithmic structuring of any system that integrates software development applied to the solution of engineering problems [2].

The UML language began to take shape in 1994, when Rumbaugh joined the Rational company founded by Booch. The objective was to unify two methods that they had developed: the Booch method and the OMT (Object Modeling tool). In 1995 he joined Rational another renowned researcher, Jacobson. These three people are known as the main managers of the development of the UML at present [3, p. 2], In addition to the aforementioned, its structural platform was oriented from the beginning to the collaboration of other companies and researchers, so that through its concepts it becomes a robust alternative for the sequential structural design of any process [4, pp. 1-6].

Industrial automation is a field of electronics and applied computing that evolves every day in its presentation with the end user, reevaluating several phases of its basic structuring process. The modeling of industrial systems is essential for the correct conformation and distribution of inputs and outputs of any process, through the correct analysis, modeling and validation of a system, you get an implementation with practical connotations consistent with the needs of the company, entity or industry [5, pp. 1-3], the variety of languages for this structuring makes it imperative to standardize this process with a matrix language, as it is presented in the current research work using the UML language, and the class diagrams as the basic architecture of the aforementioned language.

Finally, and through the present investigation, the aim is to unify the computational sciences with the industrial technical sector, through the integration of the language oriented to the modeling of systems and processes (UML), making a main emphasis in the class diagrams, to denote the need of structure prior to industrial processes, prior to its implementation and integration both logical and practical, standardizing methods for development, providing alternative solutions in this area,

optimizing classic processes based on the conception of systems, tending to standardize and implement the so-called "workflow" "In the industrial sector, which are characteristic of the processes that link the aforementioned subject, increasing the productivity of the organization [6].

2 UML as A Proposal in Automation

2.1 Definition of UML (Unified Modeling Language)

UML is a language oriented to model computer systems of great complexity and with a defined structure, also allows to model non-computer systems such as workflows, design of the structure of an organization, hardware design and automation applications [7, pp. 1-2].

2.2 Types of Structured Diagrams Enabling the UML Language

The diagrams are the graphs that describe a set of elements with their relations which offer a graphic and structural view of the system to be modeled [8].

UML offers the following diagrams for structuring the modeling of case studies, among the most important are:

- Diagram of objects, collaboration, states and activities: Modeling the dynamic behavior of the system [9].
- Components diagram: Allows the deployment and development of structures, based on a practical environment for implementing components for a system, in short it is a black box whose content is not of interest to the implementer [10].
- Diagram of use cases: Allows the structured modeling of a system through the use of cases for that purpose [9].
- Sequence diagram: This type of diagram allows the interaction of the objects that make up a system temporarily [10].
- Class diagram: Describes the design of object-oriented systems using classes, attributes and relations [11, pp. 247-249].

2.3 Class diagram

It shows a set of classes, interfaces and their relations, allowing us to describe the inputs and outputs of the process, as well as their types, and also to name the processes that involve them. This is the most common diagram when describing the design of object-oriented systems [12].

A class diagram is composed of the following elements:

- Class
- Relations
- Responsibilities

Class: It is the basic unit that encapsulates all the information of an object type (Object is an instance of the class). Through it we can model the study environment [9].

A class diagram is composed of the following elements:

- Classes: attributes, operations and visibility.
- Relations: Heritage, Composition, Aggregation, Association and Use.
- Responsibilities

2.4 Attributes

The attributes provide a detail of a specific class, these can be of three different classes:

- Public: Details that the attribute will be visible both outside and inside the class
- Private: Specifies that the attribute will be accessible only within the class
- Protected: Specifies that the attribute will not be accessible from outside the class, but will provide access to the subclasses that derive it.

2.5 Operations (Methods)

The operations detail the way in which the class relates to the environment, in the same way they can be public, private and protected.

2.6 Relations

There are generally three types of relations:

- Dependence
- Generalization
- Association

2.7 Dependence

A change in the independent class (Application) can affect the dependent class (Window), establishing a direct relation between the mentioned stages, which allows defining the concept of dependence, for the class diagrams (See Fig. 1).

Fig. 1. Dependency example [13]

2.7 Generalization

It allows to relate a general class (superclass), with a breakdown of it (subclass), a class without a superclass is called a root class, instead a class without a subclass, it is called as class leaf (See Fig. 2).

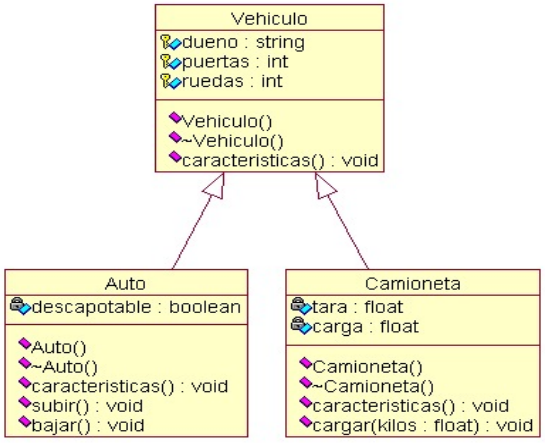

Fig. 2. Example of a superclass [13]

2.8 Responsibilities

Responsibilities are defined by identifying the classes that make up a given system, through a particular analysis, for each section presented (See Fig. 3).

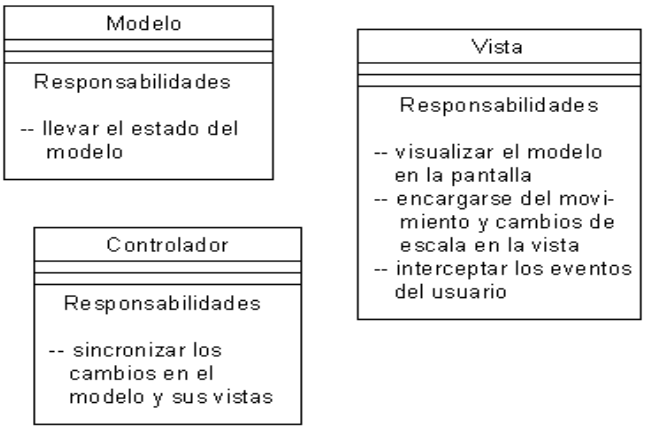

Fig. 3. Example of responsibilities in a class [13]

3 Development of the Conceptual Model based in UML

The application of UML as standard appears in front of the deficiency of some standards, among them the IEC 61499 used for the modular approach of an architecture of function blocks [14]. However, a special apparatus is required for real-time response, distribution, interfaces of the technical process and limitation of resources. The approach to solve these needs is to generate a hybrid between the use of function blocks and UML, using C ++ standards resulting in an improved IEC 61499 standard [15].

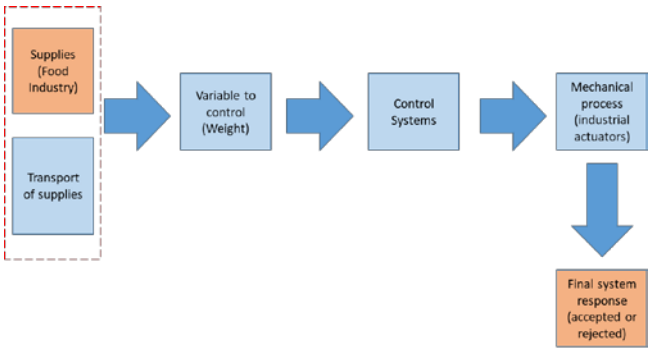

Fig. 4. Food packaging system

An example of the above is the presentation of an automated food packaging system, for the present case study the weight of the product is a determining factor, given that an established weight, derives in the two main standardized functions of the system, "use or recycle" (See Fig. 4).

The rules of domain mapping, composed of IEC 61499 and UML, allow the construction of meta-models [16], Fig. 5 shows an approximation between the implementation of the system using industrial field languages and the conceptual structuring of the model of this case study, based on UML language.

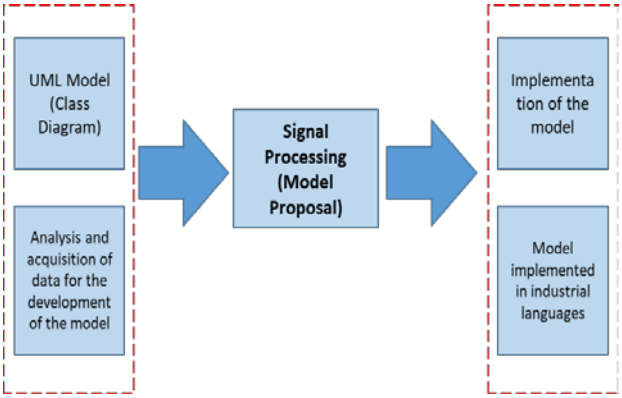

Fig. 5. Processing of UML language and field languages

4 Evaluation of the Proposed Model

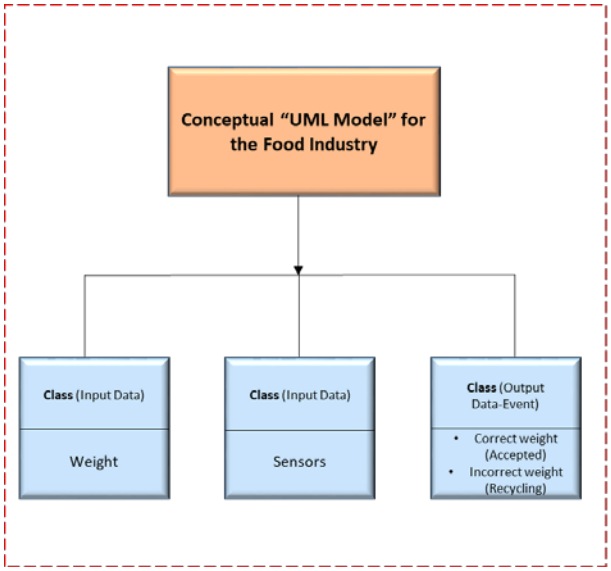

Fig. 6. Conceptual UML Model (Food Industry-Class Diagram)

For the discussion of the present research proposal, the development of the structure of a class diagram, based on the packaging of the raw material in the food industry, is presented as the main case study (Fig.6).

The previous case study exposes the facility for the integration of the UML language, specifically the class diagrams in the field of control and industrial automation, specifically the development of a packaging system for the food industry, which is a generalized case for the direct application of UML in the industrial automation, integrating and denoting the practicality of the UML Language, in processes of continuous and non-continuous production.

The strength of the UML language and class diagrams in general, is the standardization presented in object-oriented programming by this type of sequential algorithmic structure, converting classical industrial processes (programming) into innovative processes applied to any controller, actuator, sensor or device based on the principles of industrial automation and control systems.

5 Conclusion

UML is an effective tool for the modeling of automated processes, it allows to migrate in a simple way to any programmable logic technology whether they are microcontrollers or PLC's, they can be programmed in any technical language and even computer systems that use C ++, for its structuring. Class diagrams are the most used and standardized diagram types of the UML language for the industrial sector, specifically the automation and control sector. They are highly efficient in their implementation, allowing the incorporation of architectures based on the systems model, with a scope defined for field technicians and programmers.

References:

[1] L. A. Arias Barragan, «Lenguaje de modelamiento unificado (UML) para modelamiento de embotelladora,» *Scientia et Technica año XXI*, pp. 38-42, 2016.

[2] T. Quatrani, «Introduction to UML 2.0,» IBM Software Group, 2005.

[3] S. S. Alhir, «Guide to Applying the UML,» New York, Springer Verlag, 2002, pp. 1-2.

[4] E. Hernández Orallo, «El Lenguaje Unificado de Modelado,» *Universitat Politècnica de València*, pp. 1-6, 2002.

[5] R. Correa Perez y R. Gustabello Cogle, *Modelado de sistemas de automatización con el uso de "UML" y "Redes de Petri"*, Santa Clara: Universidad Central "Marta Abreu "de Las Villas , 2011.

[6] F. Zorzan, M. Daniele, M. Frutos y M. Uva, «Una herramienta para la Automatización de Procesos de Desarrollo de Software usando QVT: Transformación de Controles de Flujo SPEM a BPMN,» de *XX Congreso Argentino de Ciencias de la Computación*, Buenos Aires, 2014.

[7] M. Fowler, UML gota a gota, Masschusets, E.U.A: Wesly Longman, Inc, 1999.

[8] M. Fossati, Introducción a UML: Lenguaje para modelar objetos, Natsys, 2017.

[9] J. E. Gracia Burgués, Aprende a Modelar Aplicaciones con UML, IT Campus Academy, 2018.

[10] L. Debrauwer y F. Van Der Heyde, UML 2.5: iniciación, ejemplos y ejercicios corregidos, Barcelona, España: ENI Ediciones, 2016.

[11] S. R. Gómez Palomo y E. Moradela Gil, Aproximación a la ingeniería del software, Madrid, España: Editorial Universitaria Ramón Areces, 2015.

[12] C. Vidal Sabino, «Propuesta y Aplicación de Diagramas de Clases UML JPI,» *La Serena* ,

2014.

[13] J. Santoyo, «Diagramas de Clase,» *UNAB*, pp. 1-64, 2016.

[14] M. V. García Sanchez, *Metodologías para el diseño de sistemas de control distribuido bajo el estándar IEC 61499 aplicados al control de procesos,* Bilbao: Universidad del País Vasco, 2018.

[15] F. Bitsch, P. Göhner, F. Gutbrodt, U. Katzke y B. Vogel-Heuser, «Specification of hard real-time industrial automation systems with UML-PA,» de *3rd IEEE Int. Conf. Ind. Informatics, INDIN*, 2005.

[16] K. Thramboulidis, «Using UML in control and automation: a model driven approach,» de *Ind. Informatics, 2004. INDIN '04. 2004 2nd IEEE Int. Conf*, 2004.