

Lenguaje de modelamiento unificado (UML) para modelamiento de embotelladora

Unified Modeling Language (UML) for bottling modeling

Luis Alejandro Arias Barragan

Fundación Universidad Autónoma de Colombia, Bogotá, Colombia

lincarias@yahoo.com

Resumen— El presente artículo describe el modelamiento de un bucle de embotellado de una planta de procesos, el caso de estudio corresponde a la planta de procesos del laboratorio de automatización industrial de la Universidad Autónoma de Colombia (FUAC). Inicialmente se define que es el UML (Lenguaje de Modelamiento Unificado), luego se describen las fases de trabajo realizadas para obtener los modelos del bucle de embotellado: Los casos de Uso, los diagramas de clases; y finalmente el diagrama de secuencias que ilustra mensajes, avisos y alarmas mediante los cuales el bucle interactúa con los demás elementos de la planta de procesos.

Palabras clave— Automatización, Casos de Uso, Diagrama de clases, Diagrama de secuencias, Lenguaje de modelamiento unificado (UML), planta de procesos.

Abstract— This Paper describes the modeling of a loop Bottling plant processes, the case study corresponds to the process plant industrial automation lab at the Autonomia University of Colombia (FUAC). Initially defined to be the UML (Unified Modeling Language), the describe the phases of work done for the loop models bottling: Use Cases, Class Diagram, and finally the Sequence Diagram that illustrates messages, warnings and alarms through which the loop interacts with the other elements of the process plant.

Key Word — Automation, Class Diagram, Process Plant, Sequence Diagram, Unified Modelling Language (UML), Use Case.

I. INTRODUCCIÓN

UML se define como un lenguaje gráfico para visualizar, especificar, construir y documentar los artefactos de un sistema con gran cantidad de software. Proporciona una forma estándar de diagramar planos de un sistema, abarcando las partes conceptuales (funciones del sistema, y en principio también procesos industriales), y los objetos concretos (clases escritas en lenguajes de programación específico, esquemas de bases de datos, componentes de software reutilizables) [1].

El problema del modelamiento de sistemas automatizados a través de lenguajes de software abierto como UML se aborda en el trabajo de Nickel [2], donde se propone la integración de

diagramas UML con esquemas SDL (Lenguaje de Diagramación Específico), utilizados en ingeniería eléctrica y mecánica.

Cada vez que se quiere soportar el diseño e implementación de una solución automatizada se debe contar con la documentación adecuada tanto para el desarrollo, como para su posterior mantenimiento o eventuales modificaciones. Resulta además conveniente contar con representaciones visuales del sistema sobre el que se desea operar, ya que la visualización adecuada permite comprender mejor lo que está creando [3].

De otra parte es importante que tanto desarrolladores, como consultores, clientes y usuarios de los sistemas tengan puntos de convergencia en la interpretación estructural y funcional de los eslabones de un sistema; esto se hace posible mediante la utilización de notaciones de diseño que todos los actores involucrados en el sistema (diseñadores y usuarios), deben aceptar como pautas, de la misma forma que los planos de ingeniería se convierten en guías para los ensambladores. El UML constituye ese tipo de notación para diseño.

UML es un lenguaje de modelado universal, por lo que cada vez más, es empleado para la descripción de arquitecturas. En este caso, se usa para la descripción de la Arquitectura de Componentes Genéricos, y en particular, para especificar la estructura interna de los elementos arquitectónicos mediante los diagramas de clases y de secuencia [4].

La formalización de estos diagramas permite establecer qué es para cada uno de estos tipos de modelos, relaciones de inclusión y refinamiento y a partir de esto probar la consistencia interna de cada uno de los elementos, así como la verificación de las interconexiones entre dichos elementos.

En el artículo “Constructing an Ontology for Multi Agent System for a software Engineering Perspective: A Case Study”, los autores presentan una estrategia para la construcción de los diagramas colaborativos que estructuran diferentes jerarquías en un sistema, de manera similar a como

se estructuran las cadenas de producción en los sistemas automatizados [5].

La principal ventaja de UML es que constituye un lenguaje de propósito general, aunque esto en ocasiones puede llegar a convertirse en una desventaja, porque no se pueden representar en toda su dimensión y detalle las situaciones o características propias de dominios específicos [6].

En el artículo “Principios de una metodología para integración empresarial bajo enfoque holónico”, los autores muestran de una forma clara como a través de UML, se puede hacer el modelamiento de las instancias de gestión y supervisión de los procesos en una planta de automatización industrial, coordinando todos los elementos encaminados a dar respuesta a los requerimientos de los clientes [7].

UML es un lenguaje universal que permite a los desarrolladores centrarse en soluciones automatizadas de los aspectos más relevantes del proceso creativo, al igual que se destacan las relaciones de causa y efecto entre las variables y los parámetros de una planta, su comportamiento dinámico y su comportamiento en el tiempo. Al ver las interacciones de los elementos del sistema como vínculos que deben cumplir una función específica, pueden entonces ser identificadas para convertirlas en objetos de diseño o subrutinas que se pueden ejecutar en diferentes tecnologías como las de microcontroladores, PLC (Controladores lógicos Programables) o incluso en computadores personales[8]

El modelamiento en UML de escenarios tanto convencionales como anómalos, se le denomina Casos de Uso; la caracterización estructural y funcional de un eslabón o bucle se realiza mediante el diagrama de clases; y finalmente las interacciones entre los bucles son descritas por el diagrama de secuencias de acuerdo a la metodología UML [9].

II. Modelamiento de planta embotelladora

El bucle seleccionado para el modelamiento a través de la herramienta UML corresponde a la embotelladora perteneciente a la planta de procesos del laboratorio de automatización industrial con que cuenta la facultad de Ingeniería de la FUAC, cuyos detalles pueden observarse en la figura 1.



Figura 1. Embotelladora de planta de procesos del laboratorio de automatización industrial de la Facultad de Ingeniería de la FUAC.

La fase inicial del modelamiento se realizó el diagrama de los Casos de Uso, en otras palabras, las diferentes acciones que pueden llegar a enfrentar tanto un programador como un operador del sistema de control de la embotelladora. Para la realización de la diagramación de los Casos de Uso, diagramas de clases y de secuencia se utilizó una herramienta de software libre para modelamiento UML denominada astah. A continuación la figura 2 puede verse el diagrama de Casos de Uso realizado.

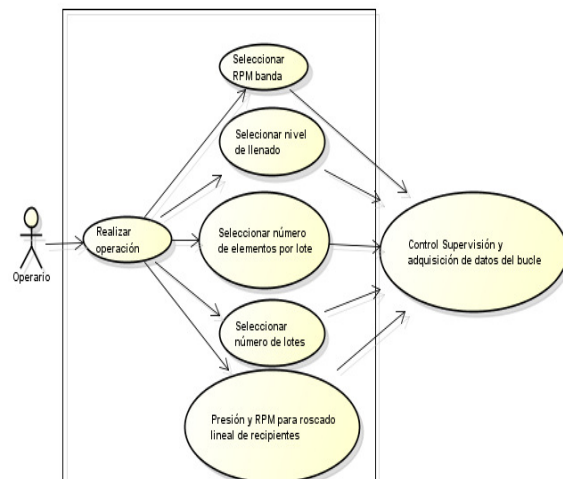


Figura 2. Diagrama de Casos de Uso para embotelladora.

En el diagrama de Casos de uso expuesto pueden verse las acciones que pueden llegar a seleccionar y configurar tanto los programadores como operarios del sistema. Entre dichas acciones se destacan: La selección de RPM (Revoluciones Por Minuto) de la banda transportadora de la embotelladora, del nivel de llenado para los recipientes que conforman un lote, el número de recipientes o envases de un lote, el número de lotes; y la presión y RPM para el roscado lineal en el proceso de tapado de los recipientes.

Una segunda fase del proceso de modelamiento correspondió a la realización del diagrama de Clase. En esta fase se realiza

una descripción de las variables de entrada y salida del proceso, sus tipos, y se nombran los procesos que las involucran. En la figura 3 puede observarse el diagrama de clase realizado para la caracterización del bucle de embotelladora.

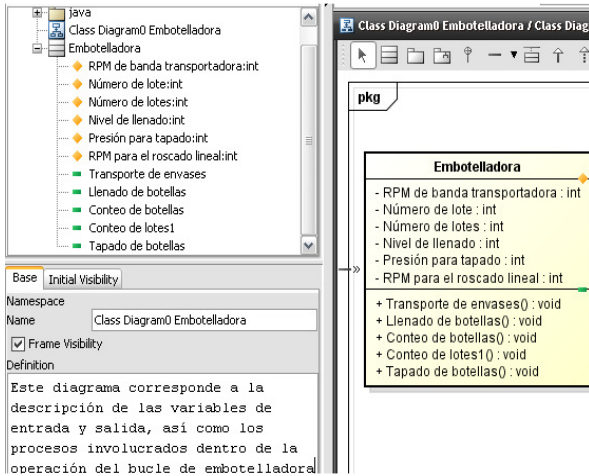


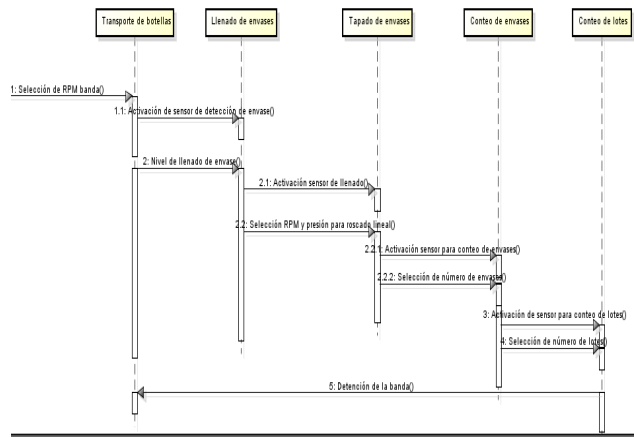
Figura 3. Configuración del diagrama de Clase para el bucle de embotelladora.

En la fase de modelamiento correspondiente al diagrama de Clases, se realiza una caracterización puntual de las variables de entrada, indicando su nombre, como por ejemplo, el número del lote, y señalando el tipo de variable para el caso concreto de tipo entero.

A las variables de entrada y salida del proceso se les denomina atributos del objeto en particular, es decir, la embotelladora tomada como caso de estudio. Las acciones que se realizan sobre dichas variables reciben el nombre de operaciones dentro de la nomenclatura UML utilizada.

La última fase realizada dentro del proceso de modelamiento por UML, correspondió al diagrama de secuencias, donde se ilustran algunos de los mensajes que son intercambiados dentro del flujo de operaciones que realiza el bucle internamente.

En la figura 4, a continuación se ilustra el diagrama de secuencia que se tiene entre las diferentes operaciones que lleva a cabo el bucle de la embotelladora.



4. Diagrama de secuencia de los procesos en el bucle de embotelladora.

En el diagrama de secuencia pueden observarse los mensajes que intercambian entre sí fases específicas dentro del proceso de embotellado, por ejemplo, se puede identificar como el mensaje o instrucción de “selección de RPM de la banda transportadora” da inicio a la fase de transporte de los envases. La fase de llenado de los envases recibe dos instrucciones: la primera del operario que indica el nivel al cual se deberán de llenar los envases, y un mensaje que proviene de un sensor indicador de llegada de los envases a la zona de llenado de los mismos.

De igual forma se detallan las demás interacciones mediante el envío de mensajes entre las diferentes fases del proceso de embotellado.

Una vez se ha realizado una descripción exhaustiva de los casos de uso, los diagramas de clase y de secuencia se procedió a través de la herramienta de modelamiento UML a obtener los esquemas básicos de las rutinas y subrutinas del proceso de embotellado en C++. La herramienta de desarrollo UML astah.6.5.1 permite realizar dicha exportación de manera simple, como se observa en la figura 5.

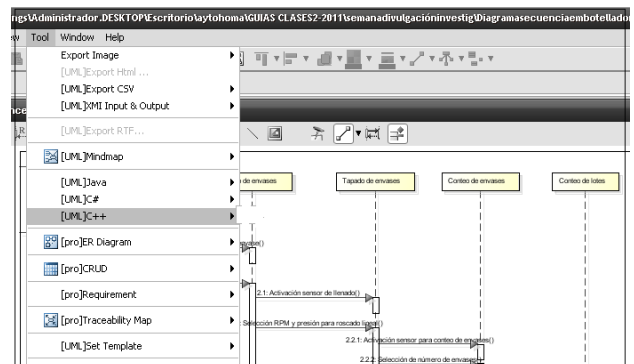


Figura 5. Exportación a C++ del diagrama de secuencia de los procesos en el bucle de embotelladora

Luego de ello se puede observar la estructura básica del programa para las subrutinas que se llevarán a cabo en C++, mismas que habrán de ser complementadas a detalle en correspondencia a la tecnología con la cual se va a trabajar, para el caso actual se decidió implementar un microcontrolador.

```
#include <string>
#include <vector>
#include <list>
#include <iostream>
#include <assert.h>

#include "Embotelladora.h"

void Embotelladora::Transporte de envases()
{
}

void Embotelladora::Llenado de botellas()
{
}

void Embotelladora::Conteo de botellas()
{
}

void Embotelladora::Conteo de lotes1()
{
}

void Embotelladora::Tapado de botellas()
{
}
```

Figura 6. Programa general en C++ que ilustra las fases del proceso de embotellado en la planta.

A continuación se desarrolla la rutina temporizada para el llenado de botellas, la cual tiene como variable de entrada la salida de un sensor de tipo capacitivo, cuya salida esta conectada al BIT 0 y produce un cambio en el B7 del puerto de un microcontrolador PIC16F877A. El Microcontrolador PIC16F877A fue utilizado por cuanto permite de una forma simple realizar su programación en lenguaje C++, mediante la herramienta de desarrollo DEV-C++ 4.9.9.2.

```
#include <16F877A.h> //LLENADO Y CONTEO DE LAS BOTELLAS EN LA EMBOTELLADORA
#fuses XT,NOVDT,PUT,NOVRT
#use delay( clock = 4000000 ) // Reloj de 4 Mhz
#use fast_io (B)

int1 cambio=0; //variable de cambio
#INT_EXT //atención a interrupción por cambio en RBO, cuando se detecta el llenado de una botella
ext_isr() {
    output_toggle(pin7); //aquí se enciende un led en el pin 7 para contar la botella llena
}

void main() {
    set_tris_B (0x01); // B0 como entrada (aquí se lee el sensor de llenado, B7 como salida
    output_low(PIN_B7); //Apaga el led del pin 7
    port_b_pullups(TRUE); // PULL-UP PARA RBO
    enable_interrupts(int_ext); // habilita int RBO
    ext_int_edge (L_TO_H); //por flanco de subida
    enable_interrupts(GLOBAL); //habilita int. general
    while (1) {
        if (bit_test(portb,0) == 1) //Si RBO es 1 apaga el LED
            bit_clear(portb,1);
        else
            bit_set(portb,1); //Si RBO=0 enciende el LED
    }
}
```

Figura 7. Programa en C++ para el PIC 16F877A que ilustra el proceso de llenado y conteo de las botellas en la embotelladora.

Debe clarificarse que la herramienta de UML proporciona tan solo la estructura general de las subrutinas modeladas y las variables a tener en cuenta, pero corresponde a los desarrolladores realizar los programas como tal en el software correspondiente a la tecnología de desarrollo seleccionada.

III. CONCLUSIONES

El UML resulta ser una herramienta eficaz para el modelamiento de procesos automatizados, brindando la oportunidad de migrar de forma simple a cualquier tecnología de lógica programable tales como Microcontroladores, PLC programado en lenguaje de listas e incluso ordenadores, que utilicen C++.

Para el modelamiento futuro considerarán algunas variables como el nivel de tanques de llenado, la temperatura del líquido, cuyo estado será tenido en cuenta para la interacción del bucle embotelladora con otros bucles cercanos.

De igual forma se contempla la “traducción” de los diagramas de clase y de secuencia a sus rutinas equivalentes en el software de programación C++ y a lenguajes de programación estructurado para PLC como el LADDER (Lenguaje escalera) y lenguaje gráfico (GRAFNET).

REFERENCIAS

- [1]. C. Parra, R. Ruiz, P. Paz, “Modelado de procesos y desarrollo de sistemas software: Integración UML y EPC”. IX Congreso de Ingeniería de Organización. Guijón, 8 y 9 de septiembre 2005.
- [2]. U. Nickel, H. Kohler, J. Niere and A. Zunday , " Integrating UML Diagrams for Control Systems " , 2000.

- [3]. G. Booch and Jacobson, *El lenguaje Unificado de Modelamiento. Manual de referencia*. Addison Wesley, Madrid 1998.
- [4]. S. Pérez, F. Orejas, N. Fuentes, “Automatización de la arquitectura de componentes genéricos usados en UML”. 2005.
- [5]. J. Paz, R. C. Bonfante, and L. Arrieta , "Constructing Ontology for a Multi Agent System for a Software Engineering Perspective: A case Study". 2010.
- [6]. G. Engels. UML – A Universal Language? ICATPN 2000, LNCS 1825, Berlin, Heidelberg, Springer-Verlag. 2000.
- [7]. E. Chacón, J. Velasco J, Rojas O. Principios de una metodología para integración empresarial bajo un enfoque holónico, 2005.
- [8]. B. Vogel-Heuser, D. Friedrich, U. Katzke and D. Witsch, “ Usability and benefits of UML for plants automation”, 2005.
- [9]. J. Gutiérrez, J. Mora, F. Escalona, J. Torres, A. Torres "Implementación de pruebas del sistema. Un caso práctico", 2007.