

Proyecto de Teoría y Laboratorio de Ingeniería de Software II

Un framework para facilitar la reutilización en la construcción de sistemas de medición

Programa de Ingeniería de Sistemas

Prof. Julio Ariel Hurtado Alegría

1. Introducción

Los sistemas de medición son cada vez más importantes para las industrias de fabricación, debido al alto nivel de automatización de los procesos de producción. Aunque la mayoría de los sistemas de medición tienen mucho en común (dominio de medición) y son costosos de construir, estos sistemas a menudo se desarrollan desde cero, y casi no se reutilizan los diseños e implementaciones disponibles. Para abordar esto, se les ha encargado a ustedes el diseño e implementación de un framework orientado a objetos y una arquitectura de componentes/servicios independientes para acelerar el desarrollo de sistemas en el dominio de la medición. Es decir, los desarrolladores partirían de un núcleo para construir sus sistemas particulares de medición. Los analistas de dominio han hecho su trabajo previo, y le han entregado un conocimiento del dominio de medición, el cual deberán usar para concretar más su diseño.

2. Requerimientos Funcionales

Los sistemas de medición son una clase de sistemas utilizados para medir los valores relevantes de un proceso o un producto. El sistema de medición se usa para el control de calidad en productos producidos que luego, se pueden usar para separar los productos inaceptables, o para categorizar los productos en función de su calidad. En algunos sistemas, los resultados de las mediciones se almacenan, en el caso de que, en el futuro, surja la necesidad requerirse esta información, p. ej. si los clientes se quejan sobre los productos que pasaron la prueba del sistema de medición.

Aunque un sistema de medición contiene cantidades considerables de software, una parte sustancial de estos sistemas es Hardware, ya que está conectado al mundo real a través de una serie de sensores (p.e. un sensor de movimiento) y actuadores (p.e. un motor). Los sensores proporcionan información sobre el mundo real a través de los impulsos leídos. Con respecto a los actuadores, a

menudo solo necesita pasarles un valor establecido expresado en conceptos de dominio de aplicación como por ejemplo velocidad angular o fuerza.

Un Sistema de Medición, sin embargo, es más que sensores y actuadores. Comienza un ciclo de medición típico con un disparador que indica que un producto, o elemento de medición, está ingresando al sistema. El **primer paso** después de este disparo viene la fase de recolección de datos por los sensores. Los sensores miden las diversas variables relevantes del elemento de medición. El **segundo paso** es de análisis durante la cual los datos de los sensores se recopilan en una representación básica y se transforman hasta que tengan el formato en el cual pueda compararse con valores ideales o umbrales. Según esta comparación, se pueden deducir ciertas discrepancias que, a su vez, conducen a una clasificación del elemento de medición (por ejemplo, productos defectuosos). El **tercer paso**, es la acción. De acuerdo a la clasificación del elemento de medición, durante este paso se realizarán las acciones asociadas a través de los actuadores. Las acciones, por ejemplo, pueden ser rechazar el elemento, lo que hace que los actuadores eliminen el elemento desde una cinta transportadora, empujándolo hacia un contenedor separado, o imprimir la clasificación en el artículo para que posteriormente pueda ser identificado automáticamente. Uno de los requisitos en el paso de análisis, es la forma en la que la transformación de información tiene lugar, las características en función de las cuales se clasifica el artículo y las acciones asociadas con cada clasificación. Éstas pueden ser simples (decisiones sobre los actuadores a través de la lectura de los sensores), pero también complejas (requerir todo un sistema de reglas de decisión).

P.

3. Requerimientos no funcionales

Intuitivo: como cualquier tipo de sistema, el framework diseñado debe basarse en conceptos que tengan una correspondencia directa en el dominio de la aplicación. La forma en que se utilizan y combinan estos conceptos debe ser lógicamente coherente con el proceso y los elementos involucrados.

Reusabilidad: el framework debe proporcionar componentes reutilizables para la construcción de sistemas de medición. Esto requiere un delicado equilibrio entre generalidad y especificidad. También significa que los componentes y las dimensiones de descomposición deben elegirse de modo que puedan componerse en piezas relativamente generales de diferentes dimensiones, para formar piezas específicas que puedan usarse en un sistema real.

Distribuido: las computaciones de sensado y de los actuadores, así como la medición y toma de decisiones se encuentran distribuidos físicamente en máquinas diferentes para soportar varios procesos de medición en una empresa

de fabricación. Por lo que se deben programar componentes independientes que faciliten la mayor posibilidad de instalación en los ambientes de producción.

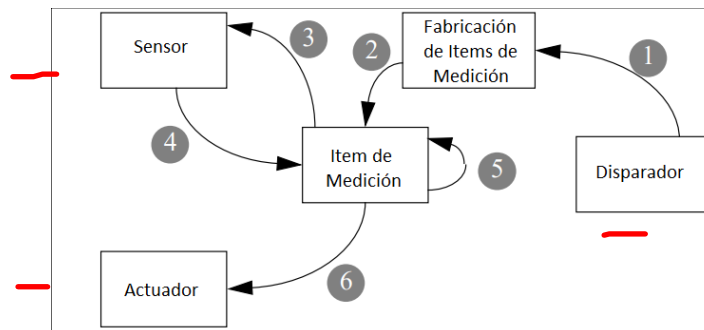
Desempeño: debido al tiempo que impone el paso de los productos por el ambiente de control de calidad, el proceso de medición debe realizarse considerando éstos límites.

Escalabilidad: es posible que una empresa quiera hacer uso del framework para desarrollar una solución multi-tenencia en varias organizaciones en un mismo sector, por ejemplo, fabricación de productos alimenticios.

Portabilidad: se busca que la aplicación sea independiente de las tecnologías hardware y los drivers por lo que se debe definir una capa abstracta y un esquema de fabricación para la instalación y uso de los dispositivos sin necesidad de modificar la solución. Para asimilar distintas alternativas se deberá pensar en una estructura de adaptadores por ejemplo para dispositivos IoT (Internet of the Things).

4. Proceso de Medición

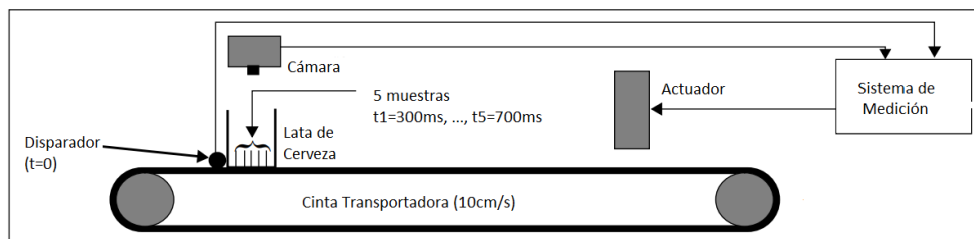
En la siguiente figura se muestra la estructura y el proceso de un sistema de medición simple, el cual consta de cinco entidades que se comunican entre sí para lograr la funcionalidad requerida.



1. El disparador activa una fábrica abstracta de ítems de Medición cuando un elemento físico ingresa al sistema.
2. La fábrica crea una representación del objeto de medición (ítem) y lo retorna.
3. El ítem de medición solicita al sensor que mida el ítem físico.
4. El sensor devuelve el resultado al ítem de medición que almacena los resultados.
5. Después de recopilar los datos requeridos, el elemento de medición compara los valores medidos con los valores ideales (valores de referencia).
6. El elemento de medición envía un mensaje al actuador para que este haga las acciones apropiadas de acuerdo a los datos medidos.

5. Aplicación Ejemplo: Sistema de medición de latas de cerveza

Al diseñar el sistema de software de lata de cerveza que utiliza el hardware que trabaja con las clases descritas (framework), se deben hacer las extensiones del framework requeridas para obtener la funcionalidad del sistema. En este caso la lata pasa por una cinta transportadora, a medida que el sistema detecta una lata de cerveza, inicia un proceso de medición que toma cinco muestras (imágenes) de la lata cada 100 milisegundos (el sensor del sistema trabaja con una cámara como sensor de imagen o un sensor de peso), que de no pasar los valores ideales (por ejemplo lata hundida o lata baja de peso), el actuador, que es un brazo, las empuja hacia la caja de cervezas que no pasaron la prueba de calidad, sacándolas del sistema de producción.



Codificación del Componente principal AppMedicionCerveza (Sólo con sensor de Peso):

```
public class AppMedicionCerveza extends AbstractApp{
    @Override
    public void crearSistemaMedicionCerveza() {
        LecturaPorEvento lector = new LecturaPorEvento();
        Disparador disparador = new Disparador(lector);
        lector.addSensor(disparador);
        FabricaItemMedicion miFabrica = new FabricaCalidadCerveza();
        disparador.setFabrica(miFabrica);
        ItemMedicion medcerveza = miFabrica.crearItemMedicion();
        medcerveza.setSensor (new SensorCamara(new LecturaSimple()));
        medcerveza.setActuador (new BrazoEmpuje());
    }
}
```

6. Simulación de un sistema aplicación

El sistema se debe probar a través de “clientes” o “publicadores” a través de una interface de usuario gráfica que permitan dar inicio al disparo y luego leer desde los sensores, que pueden ser cajas de texto u otros widgets disponibles en la tecnología de interface de usuario seleccionada. La administración de servidores,

publicadores, suscriptores debe realizarse a través de una interface gráfica de configuración cuyos valores por defecto serán leídos de un archivo de propiedades.

7. Entregas

Primer Sprint: solución aplicando y documentando (1) la aplicación del patrón de arquitectura n-niveles (cliente servidor en niveles y balanceadores de carga) del tipo componentes y conectores y haciendo uso la tecnología Java Sockets o Java RMI o equivalente y (2) el patrón de arquitectura de capas del tipo módulos para cada componente independiente de la aplicación. Se debe abstraer una capa de infraestructura o comunicaciones. Se espera el uso de al menos dos patrones de diseño orientado a objetos.

Segundo Sprint: solución aplicando y documentando (1) la aplicación del patrón de arquitectura publicador/subscriptor del tipo componentes y conectores y haciendo uso la tecnología RabbitMQ o equivalente y (2) el patrón de arquitectura micro-kernel del tipo módulos para cada subsistema independiente identificado la aplicación. Se debe abstraer una un modelo de plug-ins para la infraestructura o comunicaciones. Se espera el uso de al menos cuatro patrones de diseño orientados a objetos.

Tercer Sprint: solución aplicando y documentando (1) la aplicación del patrón de arquitectura basada en servicios (puede incluir mensajería) del tipo componentes y conectores y haciendo uso la tecnología JEE (Spring Boot o equivalente y (2) el patrón de arquitectura Hexagonal del tipo módulos para cada subsistema independiente de la aplicación. Se deben abstraer unos adaptadores para la infraestructura o comunicaciones. Se espera el uso de al menos seis patrones de diseño orientados a objetos.

Cada entrega debe incluir la documentación que se indica a continuación, el repositorio en GitHub de las aplicaciones, las pruebas unitarias usando algún framework de automatización, la información del proyecto en Jira o Trello evidenciando la aplicación de prácticas ágiles.

Documentación

1. Arquitectura

- a. Escenarios de uso del proceso de medición y configuración del sistema de medición (ítems de medición, sensores y actuadores) pueden utilizar casos de uso o historias de usuario.
 - b. Escenarios de calidad (2 escenarios consistentes con las decisiones de la arquitectura indicadas en el sprint) y selección de tácticas correspondientes.
 - c. Vista de Módulos (Siguiendo los patrones tipo módulo indicados)
 - d. Vista de Componentes y Conectores (Siguiendo los patrones tipo C&C indicados)
 - e. Vista de Despliegue (Deployment)
2. Diseño Detallado
- a. Diagrama de Secuencia para los escenarios de uso
 - b. Diagrama de Clases (Evidenciando que se tuvo en cuenta la vista de módulos)
3. Implementación prototipo (simulando sensores y actuadores en una interface de usuario gráfica) en el lenguaje y tecnologías correspondientes a su elección y a las indicadas en las entregas.