

Sistema de información visual para pasajeros de trenes

Documento de Arquitectura

Carlos Germán Carreño Romano

ccarreno@fi.uba.ar
charlieromano@gmail.com

19/04/2020

Versión 1.0

Esta obra está bajo una
Licencia Creative Commons Atribución
4.0 Internacional



Historial de cambios

Versión	Fecha	Descripción	Autor	Revisores
1.0	10/04/2020	Documento de arquitectura	Carlos Germán Carreño Romano	

Documento de arquitectura	1
Historial de cambios	2
Introducción	4
Propósito	4
Ámbito del sistema	4
Definiciones, acrónimos y abreviaturas	5
Referencias	6
Visión general del documento	6
Arquitectura	7
Patrones	7
Patrón Observar y Reaccionar (Observe and React)	7
Patrón Segmentación de Procesos (Process Pipeline)	7
Patrón HAL (Hardware Abstraction Layer)	8
Componentes	8
Descripción de componentes	10
Interfaces	11
Interfaces externas	11
Interfaces internas	11

1. Introducción

En este documento se describen aspectos de Arquitectura del software especificado en el documento ERS. Se describen detalles de alto nivel de los componentes del software, de la relación entre componentes y de la relación con el contexto.

1.1. Propósito

El objetivo principal de este trabajo es diseñar e implementar un sistema de información visual para pasajeros a bordo del tren.

Está dirigido a:

- 1.1.1. Todos los miembros del grupo de trabajo GICSAFE y SOFSE que participan de proyectos orientados a cubrir necesidades tecnológicas del sistema ferroviario argentino.
- 1.1.2. Alumnos y personal académico con intenciones de participar en proyectos de desarrollo aplicados a la industria.
- 1.1.3. Desarrolladores de software y equipamiento para trenes.

El sistema tiene impacto directo en las formaciones ferroviarias existentes que brindan servicio al pasajero todos los días.

1.2. Ámbito del sistema

Este software formará parte del sistema a desarrollar como trabajo final de la Carrera de Especialización en Sistemas Embebidos. Referencia:

CESE.Planificaciones.SistemaDeInformacionVisualParaPasajeros.

El nombre que recibe el futuro sistema es FIUBA-PIDS-2020.

El sistema debe leer datos de información al pasajero de la red interna de los trenes y presentarlos en un display LED. El sistema no se encargará de presentar los mensajes en formato de audio.

Este sistema permitirá implementar las funciones de visualización del sistema de información al pasajero existente. El sistema comercial existente es un equipamiento propietario que integra otras funciones como el sistema de audio, un CCTV usando cámaras de seguridad, entre otras.

El sistema que se especifica en este documento busca desacoplar funciones del equipamiento comercial para permitir reponer carteles que en la actualidad quedan fuera de servicio por fallas o pérdida del material original y que no pueden ser reparados.

El valor principal que aporta este sistema es contribuir con la sustitución de repuestos faltantes por medio de desarrollo. De esta manera, se busca reducir la dependencia tecnológica de la empresa con los fabricantes.

1.3. Definiciones, acrónimos y abreviaturas

TCN	Train Communication Network
WTB	Wire Train Bus
MVB	Multi-Vehicle Bus
PIDS	Passenger Information Display System
ERS	Especificación de Requerimientos de Software
REP	Trunk module
AXMe	Analog input module
DXMe	Switching value I/O module
DIMe	Switching value input module
GWMe	Vehicle control module
RCMe	Communication module
ERMe	Event recording module
HMI	Display
PIDS	Train information display system
CCTV	Closed-circuit monitoring system
DOORL/R	Door controller left (right)
SIV	Auxiliary system
VVVF	Traction system
BCU	Breaking control unit
HVAC	Air conditioning system

1.4. Referencias

[1] PDE 2020: Sistema de monitoreo y gestión de la red TCN en formaciones ferroviarias, https://drive.google.com/open?id=1gABv_NGxrLL5w94XaqDNv7iu7PSz6bxR

[2] IEC 61375-1: Electronic railway equipment – Train communication network (TCN)

[3] IEC 61375-2-1:2012 Electronic railway equipment - Train communication network (TCN) - Part

2-1: Wire Train Bus (WTB)

[4] Electronic railway equipment - Train communication network (TCN) - Part 3-1 : Multifunction Vehicle Bus (MVB) - Matériel électronique ferroviaire

[5] MVB System User's Guide, Comments and Principles of MVB Engineering, duagon

1.5. Visión general del documento

La sección 1 de este documento es introductoria; en la sección 2 se describen los patrones arquitectónicos elegidos para el diseño de alto nivel del software; en la sección 3 se presentan los diagramas de arquitectura; en la sección 4 se describen los componentes del software y en la sección 5 sus interfaces.

2. Arquitectura

2.1. Patrones

Los casos de uso del documento de especificaciones [ERS], plantean problemas de diseño a resolver que presentan similitudes. En los casos de informar al pasajero un mensaje de asistencia de forma periódica, informar la estación arribada o las estaciones cabecera a partir de algún evento que sucede en el tren, el problema a resolver implica observar variables de estado y generar un mensaje de salida en base a algún criterio. Se utilizará para estos casos el patrón Observar y Reaccionar.

El caso de uso UC-4, receptor de tramas, presenta particularidades que pueden ser resueltas con el patrón de Segmentación de Procesos (Pipeline).

El software utilizará una capa de abstracción de hardware para ser ejecutado en distintas plataformas, por lo tanto también se incorpora el patrón HAL (Hardware Abstraction Layer).

La solución que se plantea en este documento de arquitectura hace uso de estos patrones de forma combinada. A continuación se detalla cada patrón.

2.1.1. Patrón Observar y Reaccionar (Observe and React)

En la figura 1 se representa el diagrama de este patrón aplicado a los casos de uso.

- **Problema:** observar variables de proceso (velocidad y control de puertas) y enviar mensajes al pasajero cuando se encuentra en una estación cabecera, arriba a una estación o está circulando.
- **Solución:** El patrón O&R desacopla las funciones de recepción de tramas de las que generan los mensajes a través de un proceso de análisis y un proceso de despliegue.
- **Consecuencias:** El patrón genera salidas que pueden ser consumidas por otros equipos o procesos, genera alarmas y presenta información sobre un display. Se obtienen mejoras en calidades de servicio como reusabilidad, mantenibilidad, complejidad y planificabilidad de tiempos de procesamiento.

2.1.2. Patrón Segmentación de Procesos (Process Pipeline)

- **Problema:** las tramas de entrada al software tienen un formato predefinido con overhead. El software debe interpretar las tramas, extraer la información útil para procesarla y descartar el resto.

- **Solución:** el patrón proporciona independencia de procesos. Mejora la eficiencia al implementarse sobre un procesador multicore. Puede contribuir con reusabilidad, mantenibilidad, distribuibilidad.
- **Consecuencias:** Se genera un flujo de datos y los valores de las variables se alojan en un en un almacén de datos.

2.1.3. Patrón HAL (Hardware Abstraction Layer)

- **Problema:** el software debe ejecutarse en distintas plataformas y para ello hace uso de una biblioteca de funciones que permiten abstracción del hardware.
- **Solución:** el patrón proporciona portabilidad, facilita la migración entre microprocesadores.
- **Consecuencias:** Permite la utilización de interfaces estándar. Puede haber overhead y afectar la performance del sistema.

2.2. Componentes

En la figura 1 se presenta un diagrama de alto nivel de la arquitectura del software. Se pueden ver los componentes y las relaciones entre los mismos. Este diagrama responde al patrón observar y reaccionar.

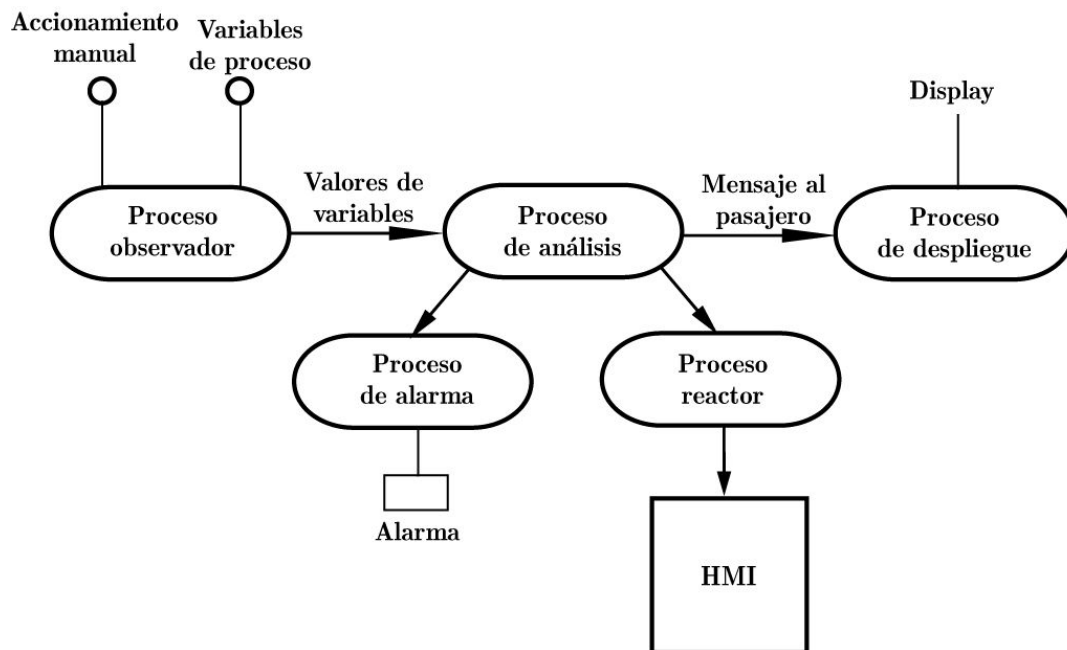


Fig. 1: Diagrama de bloques del software.

El proceso observador monitorea las variables de accionamiento manual y variables de proceso (velocidad y control de puertas). Como las variables de proceso se reciben como tramas provenientes del subsistema externo PIDS, el proceso observador es distinto respecto del accionamiento manual y se detalla en la figura 2. El proceso de análisis está compuesto por un grupo de componentes que se detallan en la figura 3. El proceso de despliegue es el encargado de generar los mensajes al pasajero y enviarlos al display. El proceso de alarma indica si hay algún error en el sistema. El proceso reactor finalmente envía datos al subsistema externo HMI.

Pipeline del Proceso observador

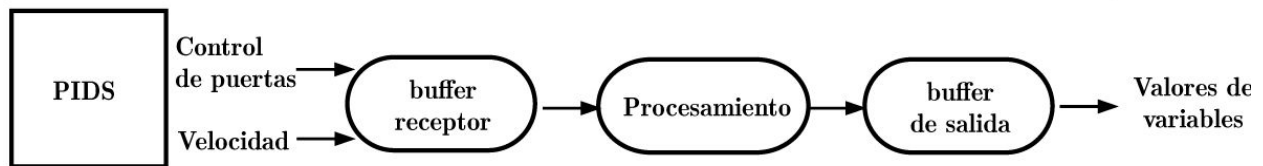


Fig. 2: Diagrama de bloques del proceso observador de variables.

El flujo de datos de las variables de proceso que provienen del PIDS sigue una segmentación de procesos. Este flujo procesa los datos recibidos en formato de tramas y obtiene a la salida los valores de las variables que serán entregadas al proceso de análisis. El proceso de análisis se detalla en la figura 3.

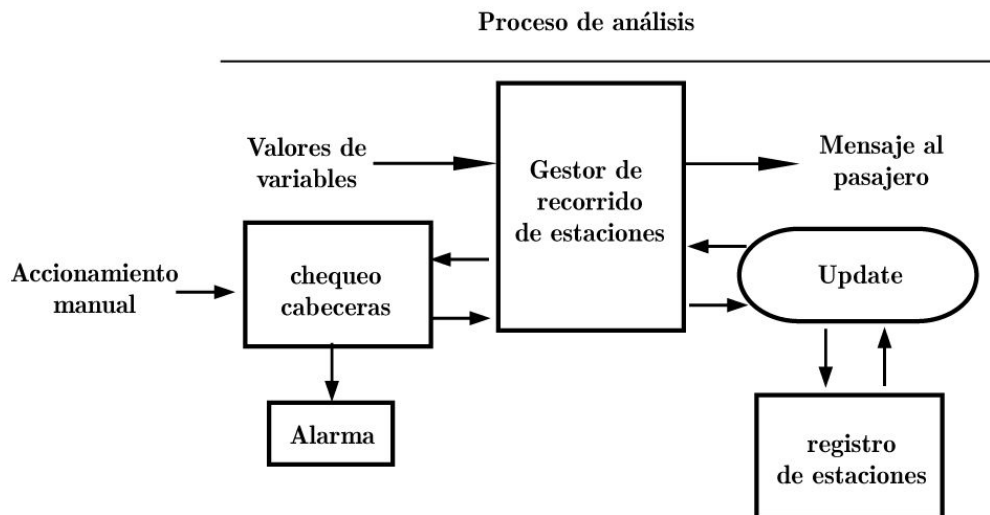


Fig.3: Diagrama de bloques del proceso de análisis

2.3. Descripción de componentes

- PIDS: sistema externo
- HMI: sistema externo
- Buffer Receptor: componente que recibe tramas provenientes del PIDS con información de las variables velocidad y control de puertas. Almacena las tramas recibidas y las envía al componente de Procesamiento.
- Procesamiento: intérprete que procesa las tramas recibidas desde el PIDS. Extrae la información de las variables y las envía a un Buffer de salida.
- Buffer de salida: almacena temporalmente los valores de las variables de proceso y las envía al proceso de Análisis.
- Proceso de análisis: se encarga de generar la orden de actualizar el sistema (cambiar de estación, enviar un anuncio al pasajero, definir las cabeceras).
- Gestor de recorrido de estaciones: es el encargado de asignar cambios en el sistema. Chequea la estación actual y asigna una nueva estación si resuelve un cambio.
- Registro de estaciones: almacenamiento estático de los recorridos con los nombres de las estaciones. Comparte información con el gestor de recorrido.
- Update: es el encargado de enviar la información de estaciones al gestor de recorridos. Recibe del gestor consultas y órdenes de actualización. Es el responsable de dar la información de la estación actual.
- Proceso observador: componente encargado de gestionar las acciones manuales del usuario para configurar las estaciones cabecera.
- Chequeo cabeceras: encargado de chequear que la estación actual sea una cabecera. Envía consultas al gestor y si recibe una estación que no es cabecera genera una alarma.
- Proceso reactor: es el encargado de generar los mensajes para el subsistema externo HMI.
- Proceso de despliegue: genera los mensajes a visualizar y los presenta en un display.

2.4. Interfaces

2.4.1. Interfaces externas

1. Interruptores: interfaz manual para que el usuario realice cambios de recorridos.
2. PIDS: subsistema de la red TCN que se encarga de la información dinámica proporcionada al pasajero, incluyendo mensajes a visualizar, mensajes en forma de audio, y monitoreo por CCTV. Entrega al software las tramas con información de variables de proceso del tren.
3. HMI: subsistema de la red TCN donde se entregan mensajes para ser visualizados por los pasajeros.

3.1.1. Interfaces internas

1. Registro de estaciones: los nombres de estaciones estarán almacenados en memoria.
2. Valores de variables: los valores de las variables de proceso del tren se obtienen del procesamiento de tramas y se entregan al gestor de recorrido. Son valores numéricos.
3. Mensaje al pasajero: los mensajes al pasajero son datos ASCII que tienen un encabezado a definir.
4. Alarmas: las alarmas se generan cuando se acciona manualmente el cambio de cabeceras durante el recorrido del tren.