

VERIFICACIÓN Y VALIDACIÓN DE SOFTWARE

Bibliografía

Fisher M. Software Verification and Validation. An Engineering and Scientific Approach. Ed. Springer. USA. 2007.

Agenda

- Verificación y Validación.
 - ✓ Análisis de la Interfaz

Verificación y Validación del software

Verificación y validación del software

3.2	Análisis de Interfaz
3.2.1	V&V debe asegurar que han sido identificados los elementos de interfaz correctos.
3.2.2	V&V debe asegurar que todos los elementos de interfaz están completamente definidos.
3.2.3	V&V debe asegurar que cada elemento de interfaz es empleado consistentemente.
3.2.4	V&V debe asegurar que los elementos de la interfaz mantienen el rendimiento necesitado por el sistema.
3.2.5	V&V debe asegurar que todos los elementos de la interfaz son testeables.

Verificación y validación del software

Tareas genéricas:

Tarea 1. Identificar los datos que deben ser transferidos entre módulos.

Tarea 2. Identificar las interfaces que manejan las transacciones de los datos.

Tarea 3. Comparar las interfaces definidas por la V&V con las de los desarrolladores y evaluar las inconsistencias.

Verificación y validación del software

Tarea 4. Analizar cada dato y determinar si se encuentra definido completamente.

Tarea 5. Graficar la ubicación de los datos y determinar si se están empleando de forma consistente.

Tarea 6. Identificar las necesidades de rendimiento del sistema.

Verificación y validación del software

Tarea 7. Modelar y simular la comunicación entre las interfaces para determinar si mantienen el rendimiento requerido.

Tarea 8. Desarrollar pruebas para las interfaces e identificar cuáles no son testeables.

Verificación y validación del software

■ Ejemplo: Proyecto MUGSEY

Objetivos

- Analiza y mantiene la salud del sistema de forma adecuada.
- Identifica y gestiona las fallas de forma adecuada.
- Adquiere, almacena y conserva los datos.
- Establece comunicación de forma confiable con la Tierra.
- Puede mantenerse fácilmente.

Verificación y validación del software

- Proveer la seguridad de que las interfaces del software soportan la identificación y gestión de fallas

Verificación y validación del software

- Los **requerimientos del sistema** que son de interés para la VyV, de acuerdo a este objetivo se definen en la siguiente tabla:

Verificación y validación del software

System Reqt. Number	System Requirement	Software Reqt. Number	Software Requirement
3.4.3.1	Stored Commanding	OS 2.4.5.1	FDIR (Fault Detection Isolation and Recovery)
6.3	Fault Handling	OS 2.1.6	Guidance Element – Update
		OS 2.1.7	Guidance Element – Invalidate
6.3.1	Science Faults	OS 2.4.2.1	Memory Manager
		OS 2.4.4	Watchdog Timer
6.3.2	Abort Mission	OS 2.4.5.1	FDIR

Planeación de la verificación y validación del software

- Proveer la seguridad de que el sistema de software identifica y gestiona las fallas de forma adecuada.
 - Se realiza una revisión de cada requerimiento del sistema y se marcan aquellos que se encuentran relacionados con el objetivo.

Planeación de la verificación y validación del software

- Para los requerimientos del sistema MUGSEY se determinó que los siguientes requerimientos están relacionados con el segundo objetivo:
 - ✓ Requerimiento 3.4.3.1 – **Stored Commanding**. MUGSEY 0x01 debe procesar una secuencia de comandos al ingresar un escenario de falla.
 - ✓ Requerimiento 6.3 – **Fault Handling**. MUGSEY 0x01 debe reconocer cuando sus subsistemas no están respondiendo y recuperarlos en 5 minutos.

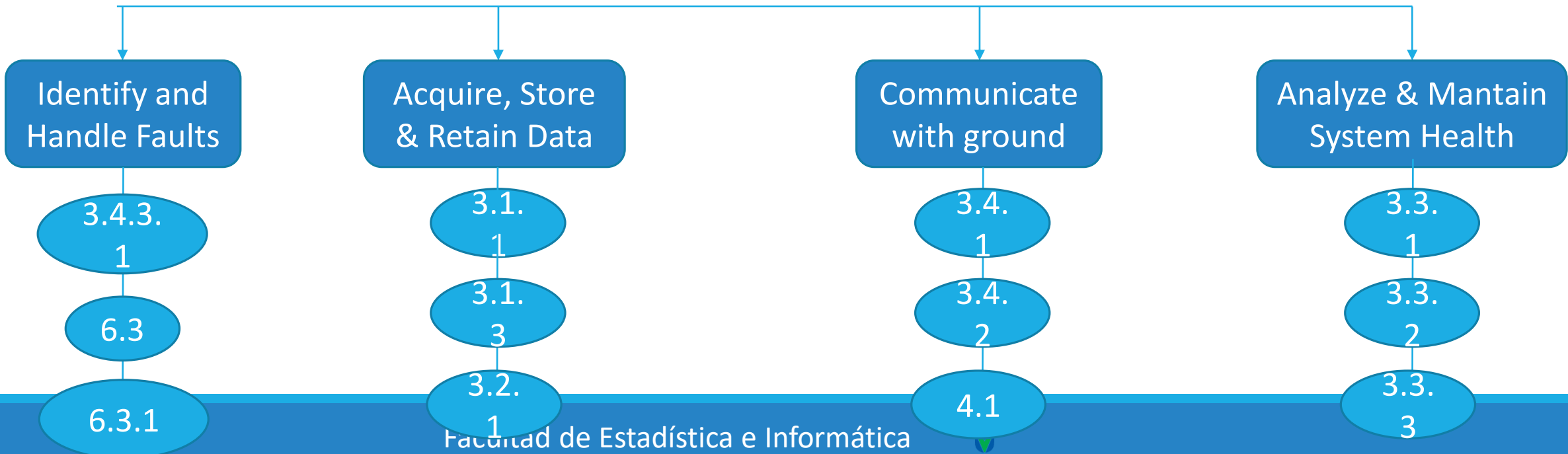
Planeación de la verificación y validación del software

- Para los requerimientos del sistema MUGSEY se determinó que los siguientes requerimientos están relacionados con el segundo objetivo:
 - ✓ Requerimiento 6.3.1 – **Science Faults**. Ninguna falla en el sistema evitará que se tomen las lecturas científicas (image, temperatura, altitud, posición) durante no más de 5 minutos.
 - ✓ Requerimiento 6.3.2 – **Abort Mission**. MUGSEY 0x01 debe reconocer cuando entra a una zona peligrosa y abortar la misión.

V&V Goal

The goal of the V&V Project is to assure that the software maintains the system's health, acquires and maintains operational data and is adequately developed to efficiently enable future missions.

V&V Objectives



Verificación y validación del software

- Por ejemplo, *Stored commanding* establece “MUGSEY debe procesar una secuencia de comandos al ingresar a un escenario de falla”. Esto ha sido asignado al requerimiento de software FDIR (Fault Detection Isolation and Recovery).
- El requerimiento de software específicamente es el OS 2.4.5.1, que establece “Si el observatorio detecta una tasa de descenso de un pie por segundo, debe enviar una secuencia de comando de RECUPERACIÓN”

Verificación y validación del software

Tarea 1. Identificar los datos que deben ser transferidos entre módulos.

Tarea 2. Identificar las interfaces que manejan las transacciones de los datos.

- **Las tareas 1 y 2** Requieren que el equipo VyV identifique los datos y las interfaces necesarias para el sistema y las necesidades de los usuarios.
- Para identificar los datos que requieren ser transferidos entre módulos, primero se identifican las fallas potenciales del sistema.
- Y en la segunda tarea se identifican las interfaces necesarias para detectar y gestionar estas fallas potenciales.

Verificación y validación del software

■ Defectos o fallas potenciales detectados:

1. El Sistema de Posicionamiento Global deja de enviar datos de posicionamiento y el sistema no lo detecta, por lo que sigue enviando la última lectura.
2. Los sensores de temperatura dejan de enviar lecturas de temperatura, el sistema no lo detecta, por lo que sigue enviando la última lectura.
3. El “Elemento orientador” deja de enviar datos de posicionamiento, pero actualiza el temporizador de vigilancia, por lo que el sistema no lo detecta y sigue enviando la última lectura.
4. La altitud disminuye o sigue siendo la misma debido a los vientos en el vehículo y el sistema piensa que está descendiendo.
5. El observatorio no reconoce cuando está descendiendo.

Verificación y validación del software

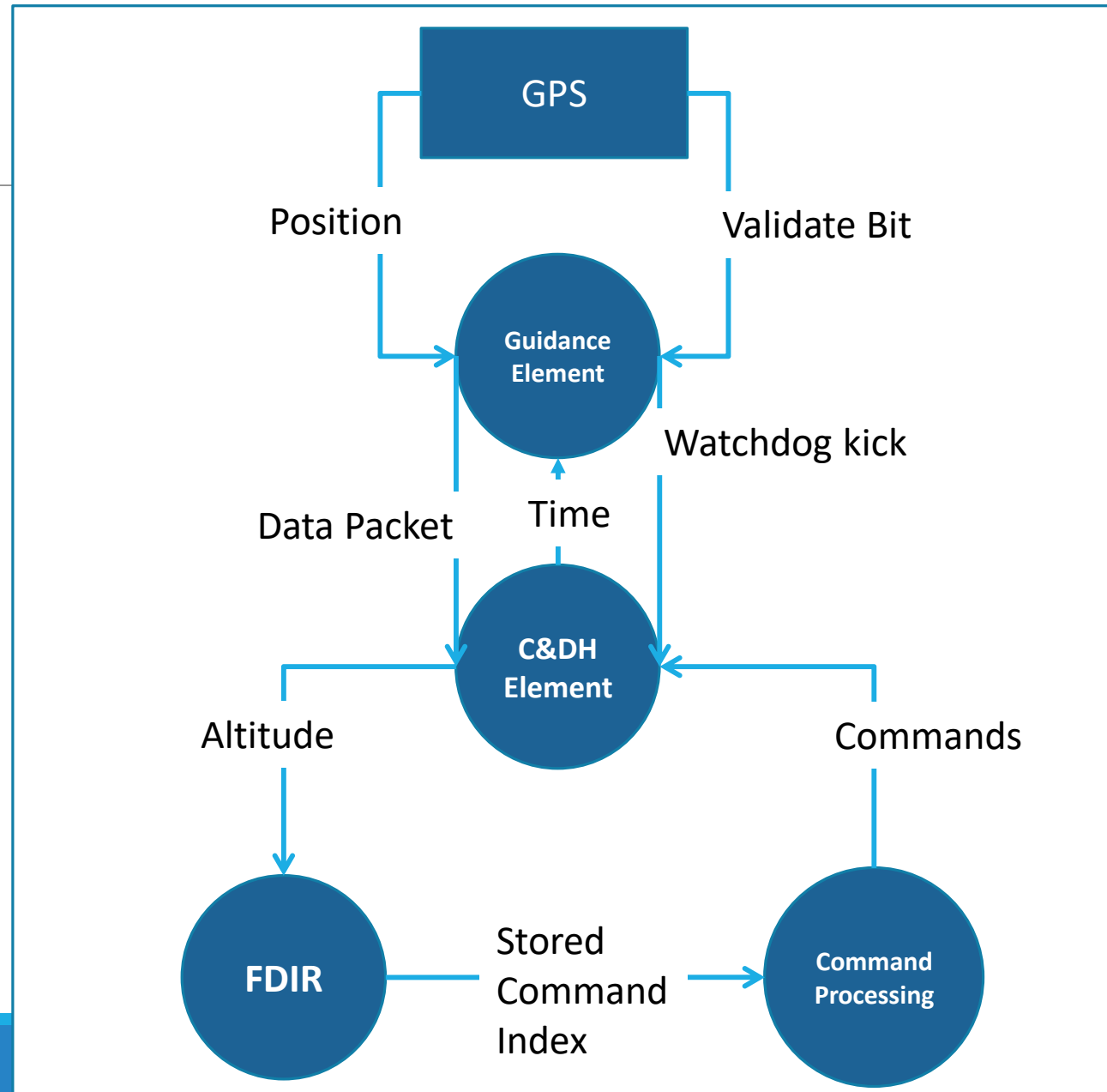
- Empleando este pequeño conjunto de fallas potenciales, que el equipo VyV obtuvo en una lluvia de ideas, se desarrolla un diagrama de flujo de datos para representar los datos que deben ser transferidos para identificar o gestionar estas fallas.
- El siguiente diagrama muestra el primer escenario de falla, propuesto por el equipo VyV:

El Sistema de Posicionamiento Global deja de enviar datos de posicionamiento y el sistema no lo detecta, por lo que sigue enviando la última lectura.

Verificación y validación del software

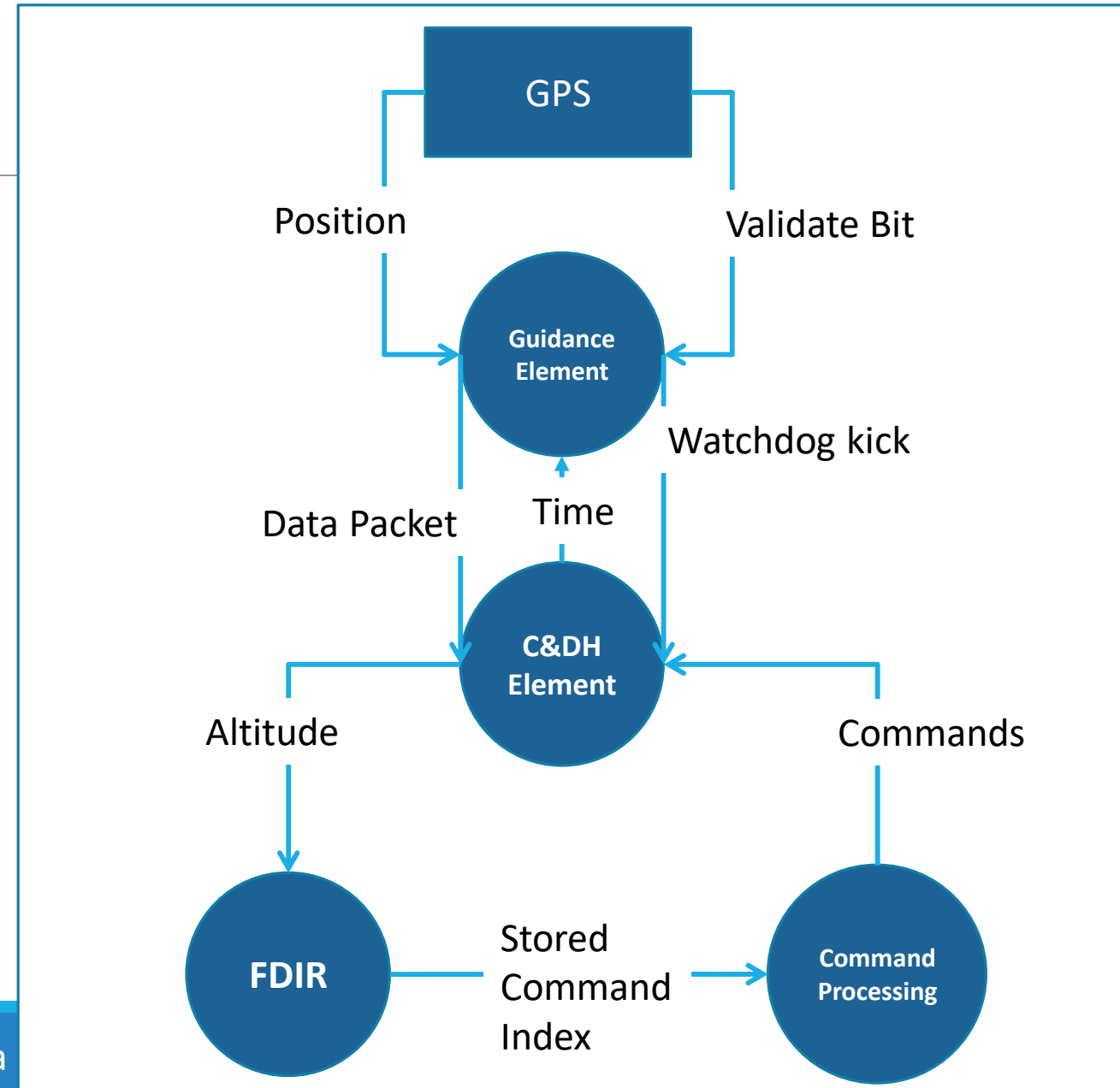
Ejemplo de diagrama de flujo de datos usado durante el análisis de interfaz.

Representa los datos que el equipo VyV piensa deberían ser representados en el sistema para manejar una falla con la unidad GPS



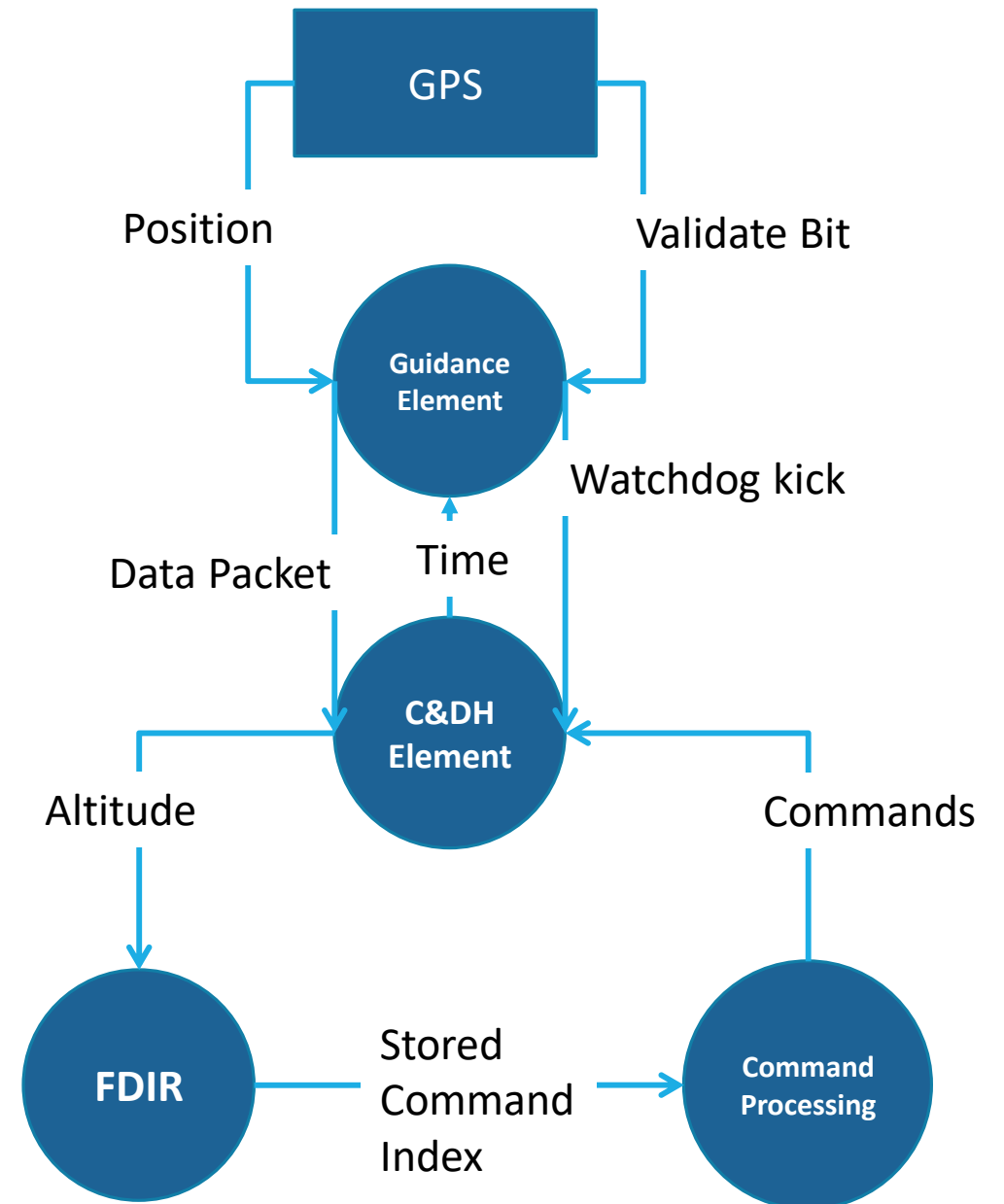
Verificación y validación del software

- Este diagrama de flujo solo encapsula la comunicación en los requerimientos 3.4.3.1 Stored Commanding y OS 2.4.5.1 FDIR (Fault Detection Isolation and Recovery)



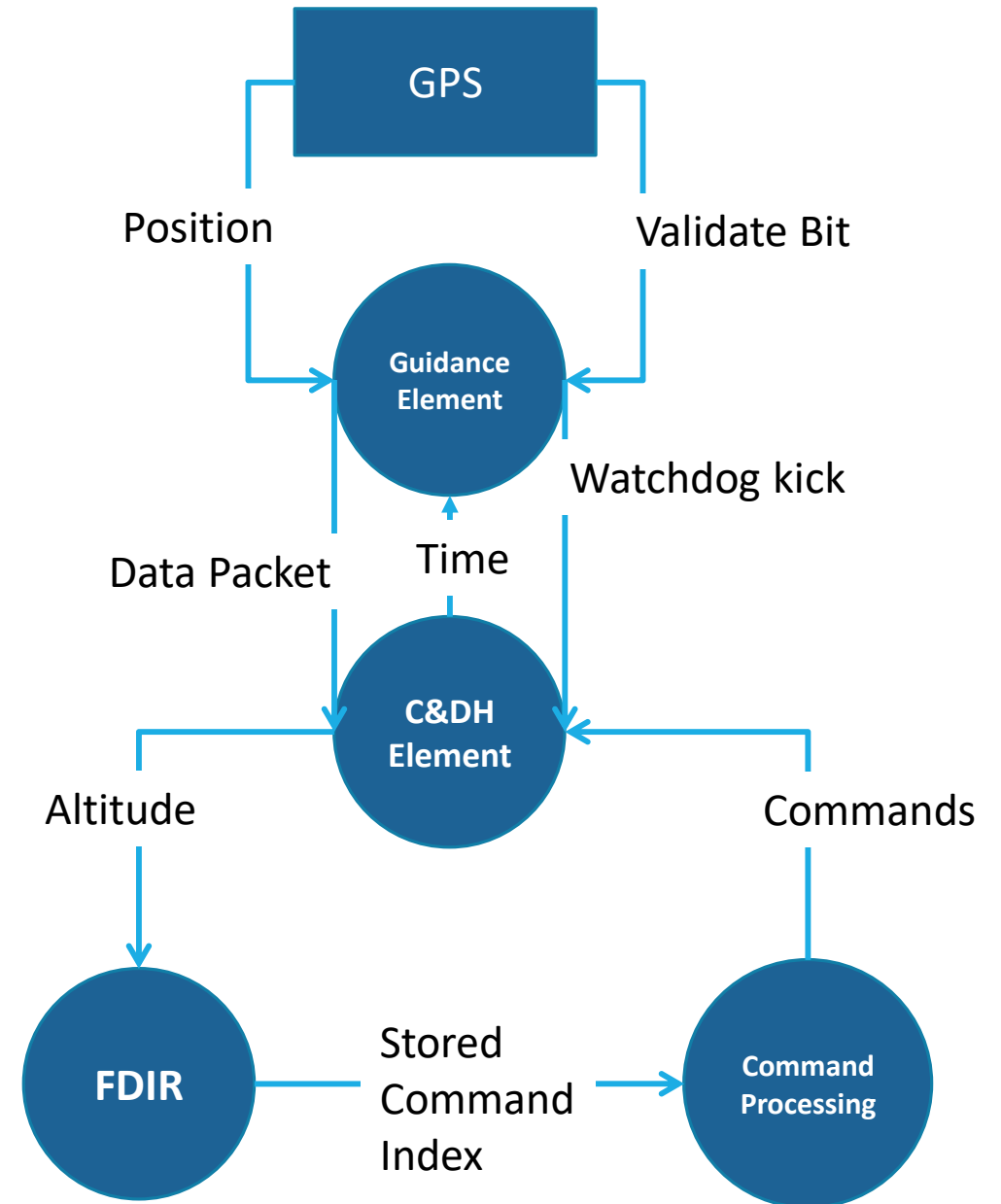
Verificación y validación del software

- La unidad GPS debe proveer datos de posicionamiento y un bit de validación al Elemento de orientación. El bit de validación sería usado para indicar si la unidad GPS está respondiendo o no. La lógica sería muy similar a un temporizador de vigilancia en el que el GPS actualiza el bit cada determinados segundos para hacerle saber al sistema que está operando normalmente.



Verificación y validación del software

- Si el bit de validación no es enviado, el sistema sabría que tiene datos inválidos y no los usa.
- El equipo VyV espera que el subsistema de recuperación, aislamiento y detección de fallas (FDIR) reciba datos de altitud a través de la interfaz y en respuesta envíe un índice al subsistema de procesamiento de comandos. El índice que es enviado a través de la interfaz sería un valor referencia para la ejecución de una secuencia específica de comandos.



Verificación y validación del software

- Existen 5 datos de interés para la VyV en este ejemplo:

Data Item	Sent From	Sent Where
Position	GPS	Guidance Element
Validate Bit	GPS	Guidance Element
Altitude	C&DH Element	FDIR
Stored Command Index	FDIR	Command Processing
Commands	Command Processing	C&DH Element

Verificación y validación del software

- El equipo VyV debe usar esta información para comparar contra las interfaces generadas por los desarrolladores. En el siguiente diagrama se muestra el diagrama de flujo de los desarrolladores:

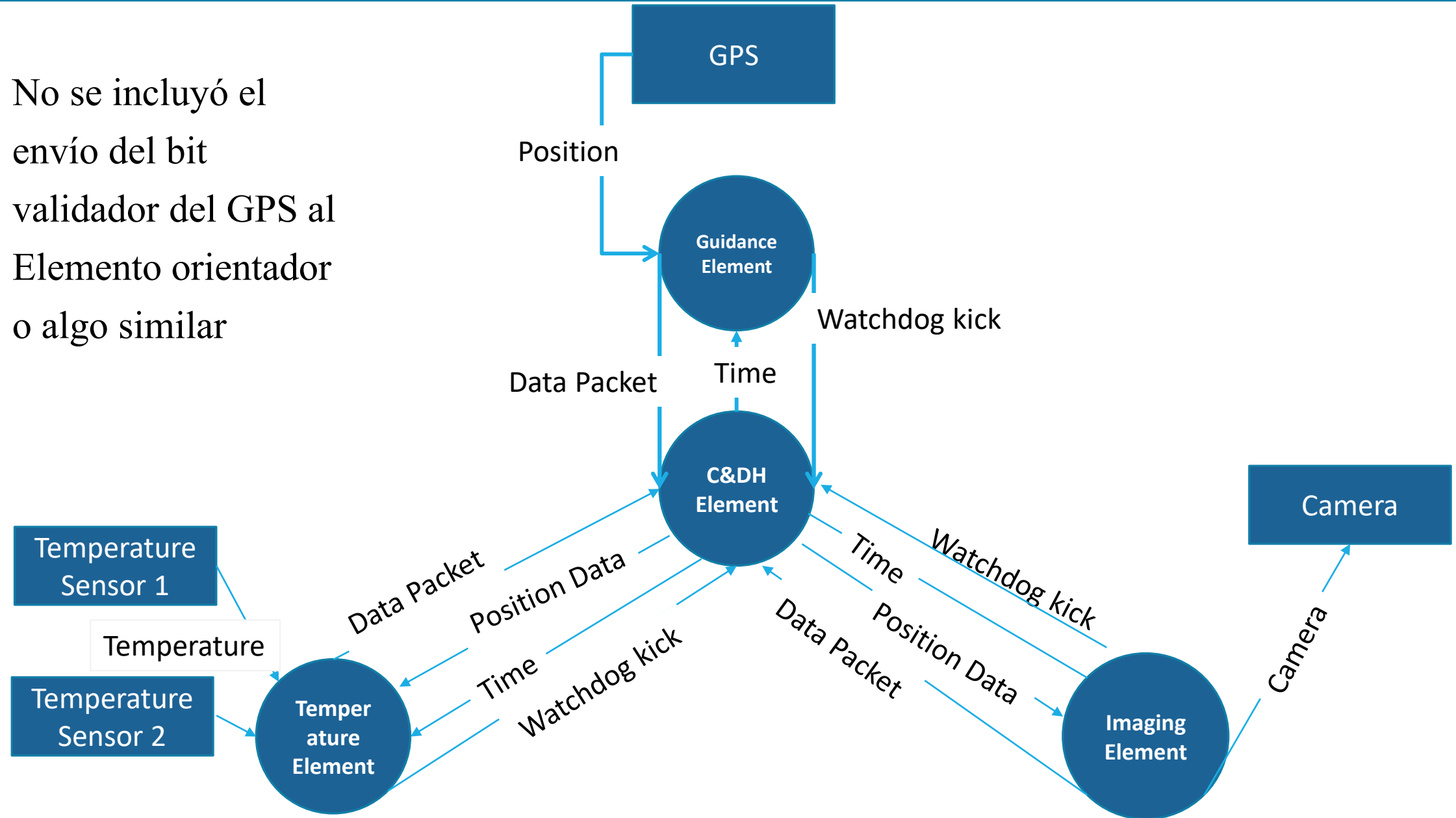
Verificación y validación del software

La Tarea 3. Comparar las interfaces definidas por la V&V con las de los desarrolladores y evaluar las inconsistencias.

Es ejecutada para comparar lo que el equipo VyV supone que debe ser definido (en cuanto a datos y su transferencia), contra lo desarrollado.

Verificación y validación del software

No se incluyó el envío del bit validador del GPS al Elemento orientador o algo similar



GRACIAS POR SU ATENCIÓN
