(19) OFICINA ESPAÑOLA DE PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 412 004**

(51) Int. CI.:

G06F 15/16 (2006.01)
H04L 12/40 (2006.01)

(12) TRADUCCIÓN DE PATENTE EUROPEA T3

(96) Fecha de presentación y número de la solicitud europea: **03.03.2006 E 06727283 (1)**

(97) Fecha y número de publicación de la concesión europea: **20.02.2013 EP 1864225**

(54) Título: **Método y producto de software para gestionar intercambio de datos en un sistema de seguridad crítico muy dinámico**

(30) Prioridad:

04.03.2005 EP 05425123

(45) Fecha de publicación y mención en BOPI de la traducción de la patente:
09.07.2013

(73) Titular/es:

SELEX ES S.P.A. (100.0%)
Via Piemonte 60
Roma, IT

(72) Inventor/es:

RINALDI, PIER PAOLO

(74) Agente/Representante:

CARVAJAL Y URQUIJO, Isabel

DESCRIPCIÓN

Método y producto de software para gestionar intercambio de datos en un sistema de seguridad crítico muy dinámico

Campo técnico

La presente invención se relaciona de manera general con el manejo de intercambio de datos en un sistema muy dinámico, a saber un sistema donde se requieren tiempos de ciclos de procesamiento de datos de menos de diezmilésimas de un segundo, que opera en un contexto crítico para seguridad de propiedad y por encima de todo para la seguridad personal, particularmente en un contexto donde están presentes diversas clases de dispositivos electrónicos, tales como, por ejemplo, ordenadores y equipos electrónicos para accionamiento, medición y control, visualización y monitoreo, y en los que se cargan aplicaciones de software, las fallas de los cuales, adicionalmente podrían producir daño considerable a la propiedad, también ponen en riesgo la vida humana.

Más en detalle, la presente invención se relaciona con un método y un producto de software para el manejo de utilización de datos entre los módulos de software productores de datos y los módulos de software consumidores de datos en un sistema muy dinámico que opera en un contexto crítico para seguridad personal y de propiedad.

La presente invención puede encontrar aplicación útil en innumerables sectores tecnológicos de los cuales, solamente por vía de ejemplo no limitativo, se puede mencionar el campo aeronáutico, y más específicamente sistemas de aviónica para aeronaves, el campo ferroviario, y más específicamente sistemas de manejo y control para trenes eléctricos de alta velocidad, en náutica, y más específicamente sistemas de manejo y control de hidroplanos, el campo de las centrales nucleares, y más específicamente el sistema de control para el núcleo del reactor, etc.

Técnica antecedente

Como se conoce, los sistemas críticos de seguridad con altas dinámicas, en general, pueden incluir una pluralidad de aparatos electrónicos, tales como sensores y accionadores, y un sistema de control central, que comprenden a su vez una pluralidad de dispositivos de interfaz hombre-máquina (HMI) a través de los cuales un usuario, por ejemplo un operador de la plataforma de referencia (un piloto de la aeronave en el caso de una plataforma de avión), puede interactuar con los equipos electrónicos, por ejemplo, para hacer selecciones o emitir comandos, por medio de un ordenador de control central a los equipos electrónicos y las interfaces hombre-máquina a través de un bus de comunicaciones.

Los aparatos electrónicos, tales como sensores y accionadores, y los dispositivos de interfaz de hombre-máquina intercambian datos a través de una aplicación de software, que se carga sobre el sistema de control central e implementa una relación directa de uso entre los módulos de gestión de software asociados con los dispositivos de interfaz de hombre-máquina y los módulos de gestión de software asociados con los aparatos correspondientes.

Una de las principales limitaciones de este tipo de mecanismo de intercambio de datos entre los módulos de gestión de software asociados con los dispositivos de interfaz de hombre-máquina y los módulos de gestión de software asociados con los aparatos electrónicos radica en el manejo de los accesos concurrentes y contradictorios a los datos del mismo módulo de gestión de software asociado con un aparato electrónico por diferentes módulos de gestión de software asociados con los respectivos dispositivos de interfaz hombre-máquina. Este problema se resuelve actualmente mediante el uso de técnicas de tipo de semáforo, a través de las cuales el acceso a los datos de un mismo módulo de software de gestión, asociado con un aparato electrónico, se habilitan a más de uno de los módulos de software de gestión, asociados con sus respectivos dispositivos de interferencia humana, sobre la base de pre-establecer prioridades.

El principal inconveniente inherente en utilizar relaciones directas de uso entre módulos de gestión de software asociados con los dispositivos de interfaz de hombre-máquina y los módulos de gestión de software asociados con los aparatos electrónicos reside en el hecho de que, en el caso que se agregue un nuevo aparato electrónico o un nuevo dispositivo de interfaz hombre-máquina, o incluso cuando se actualizan, es necesario tomar acción sobre ambas relaciones de uso que involucran estos módulos de gestión de software y sobre el manejo de acceso concurrente a los datos, haciendo así la aplicación de software insuficientemente flexible y haciendo los tiempos de desarrollo, validación y certificación de los aspectos críticos de seguridad extremadamente largos y onerosos.

En el campo de sistemas que se operan en contextos críticos de seguridad, se hacen incluso más complejos por los requerimientos solicitados para las aplicaciones en plataformas muy dinámicas, se percibe la necesidad de la creación de una arquitectura de software que permita que se logren los siguientes objetos de diseño:

• capacidad de implementar una pluralidad de dispositivos de interfaz de hombre-máquina y una variedad de sensores/accionadores a través de una abertura y configuración modular, donde el número de dispositivos de interfaz de hombre-máquina y el número de sensores/accionadores son funciones del nivel de seguridad, y por lo tanto del nivel de redundancia, solicitado por la plataforma bajo el desarrollo,

• comunicación entre una pluralidad de dispositivos de interfaz de hombre-máquina y sensores/accionadores que se logra a través de una pluralidad de instancias de una clase de software igual, únicamente definido,

• desacoplamiento de la arquitectura de software entre la pluralidad de dispositivos de interfaz de hombre-máquina y la pluralidad de sensores/accionadores que permite alta capacidad de mantenimiento de la aplicación de software y facilidad de expansión,

• capacidad de resolver posibles conflictos de acceso para las estructuras de datos compartidos, que se pueden lograr a través de una matriz de regla/prioridad, y

• una aplicación de software de conformidad con los requerimientos de procesos necesarios para apoyar los procesos de certificación asociados con aplicación de software crítico de seguridad, de acuerdo con el estándar RTCA-D0178B en el caso específico de una aeronave.

F. BUSCHMANN, R. MEUNIER, H. ROHNERT, P. SOMMERLAD, M. STAHL: “Pattern-Oriented Software Architecture Vol. 1 A System of Patterns” 30 de septiembre 1996 (1996-09-30), JOHN WILEY & SONS, GREAT BRITAIN, describe una arquitectura de software orientada a patrón diseñada para aplicaciones que necesitan interfaces de usuario flexibles y extensibles, o aplicaciones que proporcionan servicios relacionados con la ejecución de las funciones de usuario, tales como programar o deshacer. Se propone una solución en donde un patrón de diseño de Procesador de Comando separa una solicitud para un servicio desde su ejecución. Un componente de procesador de comando gestiona la solicitud como objetos separados, programa su ejecución, y proporciona servicios adicionales tales como el almacenamiento de objetos de solicitud para deshacerlos más adelante.

D. SCHMIDT, M. STAHL, H. ROHNERT, F. BUSCHMANN: “Pattern-Oriented Software Architecture Vol. 2 Patterns for Concurrent and Networked Objects” 31 de enero 2000 (2000-01-31), JOHN WILEY & SONS, GREAT BRITAIN describe una arquitectura de software orientada a patrón diseñada para aplicaciones que contienen objetos cuyos métodos se solicitan simultáneamente mediante múltiples subprocesos de cliente y a menudo modifican el estado de sus objetos, en donde para dichas ejecutar correctamente aplicaciones simultáneas, es necesario sincronizar y programar el acceso a los objetos. Se propone una solución en donde un patrón de diseño de Objeto de Monitor actualmente sincroniza la ejecución del método para asegurar que solo un método en un momento se ejecuta dentro de un objeto. También permite que los métodos del objeto programen de forma cooperativa sus secuencias de ejecución.

Descripción de la invención

El propósito de la presente invención es proporcionar una arquitectura de software para gestión de intercambio de datos en un sistema muy dinámico, a saber un sistema donde se requieren tiempos de ciclos de procesamiento de datos de menos de diezmilésimas de un segundo, que operen en un contexto crítico para seguridad personal y de propiedad, que permite que los inconvenientes de los sistemas conocidos se superen de manera general, por lo menos en parte, y, más específicamente, para alcanzar los objetivos de diseño indicados anteriormente.

De acuerdo con la presente invención se proporcionan un sistema de seguridad crítico muy dinámico y un producto de software para gestionar intercambio de datos en el sistema de seguridad crítico muy dinámico, como se define en las reivindicaciones adjuntas.

Breve descripción de los dibujos

Por motivos de simplicidad descriptiva y sin pérdida de generalidad, la presente invención ahora se describirá con referencia a una de sus innumerables aplicaciones, en particular la aplicación aeronáutica, y con referencia a los dibujos adjuntos, que ilustran un ejemplo no limitante de la realización, donde:

• La Figura 1 muestra un diagrama de bloque de un aparato de control de misión de una aeronave,

• La Figura 2 muestra la arquitectura de un ordenador de misión que forma parte del aparato de control de misión de la Figura 1, y

• La Figura 3 muestra la arquitectura con base en módulo de un software que se ejecuta en el ordenador de misión de la Figura 2 e implementa el método de gestión de acuerdo con la invención.

Mejor modo de llevar a cabo la invención

Ilustrado de manera esquemática en La Figura 1, y designado como u todo por 1, hay un sistema de aviónica de una aeronave 2 (representado esquemáticamente por una línea discontinua), por ejemplo una aeronave de entrenamiento avanzado.

El sistema de aviónica 1 comprende una pluralidad de aparatos a bordo 3, tales como, por ejemplo, sensores de identificación de navegación de comunicación (CNIs), sensores de video (infrarrojo de barrido frontal (FLIR), radar (RDR), receptor de alerta de radar (RWR), etc.), sistemas de armas (misiles aire a tierra (AGM), etc.), interfaz y unidad de cálculo de reversión (MIScellanea COmputer - MISCO), etc., y un sistema central de misión 4 que comprende un ordenador de misión (MCSG) 5, un sistema de transferencia de datos (DTS) 6, conectado al ordenador de misión 5 a través de un cable Ethernet para transferir bases de datos de la misión y registrar los datos de vuelo, y una pluralidad de dispositivos de interfaz 7 conectados al ordenador de misión 5 y a los aparatos a bordo 3 a través de un bus externo 8 del tipo MIL-STD-1553B para permitir que un usuario, por ejemplo el piloto, interactúe con los aparatos a bordo 3.

Los dispositivos de interfaz 7 comprenden una pluralidad de pantallas multifunción a color inteligentes (SMFDs) 9 y unidades de colimador de pilotaje (HUDs) 10.

En particular, en la aeronave de entrenamiento avanzado, las pantallas multifunción a color inteligentes 9 son preferiblemente seis, tres para la cabina delantera y tres para la cabina posterior, son del tipo de pantalla de cristal líquido de matriz activa (AMLCD), se proporcionan con un teclado 11 para entrada de datos o para hacer selecciones, y tiene un área de pantalla de 5” x 5”.

Las unidades de colimador de pilotaje 10 pueden ser dos, una para la cabina delantera y otra para la cabina posterior, y cada una comprenden una unidad de pantalla de piloto (PDU) 12 y un panel de control inicial (UFCP) 13.

La Figura 2 ilustra la arquitectura del ordenador de misión 5, que comprende una unidad de suministro de energía (PSU) 14, una unidad de procesamiento (PPC4-AL) 15 con base en un microprocesador Motorola Power PC750, una unidad de comunicaciones (COMMBC) 16, que se interconecta con el bus externo 8, una unidad de generación de mapas digital (SBM) 17, una unidad de control gráfico 18 del tipo de trama-vectorial (EGCRS), diseñada para generar símbolos gráficos para las unidades de colimador de pilotaje 10, una unidad de interfaz HOTAS (Manos en Mando de Gases y Palanca de Control), una unidad de selección de video (VRM) 20, que se diseña para recibir señales desde la unidad digital de generación de mapas 17, desde la unidad de control gráfico 18 y desde la unidad de interfaz HOTAS 19, y un bus de comunicación interna 21 compartido entre todas las unidades del ordenador de misión 5 para el intercambio de datos.

Cargada en la unidad de procesamiento 15 hay un programa de operación de vuelo (OFP) 22, con base en una arquitectura basada en módulos ilustrada en la Figura 3 y compilada en un lenguaje de programación conocido como Ada 95, que se basa en el uso de la construcción del “tipo protegido” que garantiza un acceso atómico a los datos y soluciona inherentemente el problema de protección de los accesos concurrentes en la lectura y escritura mediante procesos paralelos.

En particular, con referencia a la Figura 3, el programa de operación de vuelo 22 se puede dividir de forma conceptual en los siguientes objetos de software:

• un objeto de software, en lo sucesivo denominado, por razones de conveniencia, con el nombre de interfaz hombre- máquina 23, diseñada para visualización de los dispositivos de interfaz 7;

• un objeto de software, en lo sucesivo denominado con el nombre de interfaz de aparato 24, diseñado para visualización de los aparatos a bordo 3;

• un objeto de software, en lo sucesivo denominado, por razones de conveniencia, con el nombre de base de datos compartida 25, diseñada para el almacenamiento de datos compartidos entre la interfaz hombre- máquina 23 y la interfaz de aparato 24, tal como parámetros de aeronave y primaria de vuelo 2, estados operacionales del sistema de aviónica 1, y comandos dirigidos a los aparatos a bordo 3 y generados de acuerdo con las selecciones hechas por el usuario;

• un objeto de software, en lo sucesivo denominado, por razones de conveniencia, con el nombre de controlador 26, diseñado para el manejo de intercambio de datos entre la interfaz hombre- máquina 23 y la interfaz de aparato 24;

• un objeto de software, en lo sucesivo denominado, por razones de conveniencia, con el nombre de navegador 27, diseñado para la ejecución de cálculos y algoritmos durante las varias etapas de navegación, de una manera conocida por sí misma y por lo tanto no descrita en detalle; y

• un objeto de software, en lo sucesivo denominado, por razones de conveniencia, con el nombre de programador 28, diseñado para programar las operaciones ejecutadas por los diversos objetos de software, de acuerdo con una secuencia lógica descrita en lo que sigue.

La interfaz hombre- máquina 23 comprende una pluralidad de módulos 29 para gestionar los dispositivos de interfaz 7, y un módulo 30 para gestionar la selección hecha por el usuario. Cada módulo de gestión 29 se asocia a un dispositivo de interfaz respectivo 7 y crea una comunicación entre el dispositivo de interfaz 7 en sí mismo y la base de datos compartida 25.

En particular, cada módulo de gestión 29 se diseña para: adquirir una selección 31 hecha por el usuario a través del teclado 11 del dispositivo de interfaz correspondiente 7, cuya selección 31 se puede constituir por una selección adecuada a un punto de referencia en un menú presentado al usuario o también por la entrada del punto de referencia en sí mismo; exhibir en el dispositivo de interfaz correspondiente 7 símbolos gráficos que representan la selección 31 hecha; y enviar la selección 31 hecha al módulo de gestión 30, que tiene la función de recopilar todas las selecciones 31 hechas por el usuario a través de los diversos dispositivos de interfaz 7 y de resolver los posibles accesos concurrentes, contradictorios o no, a los aparatos a bordo 3.

El controlador 26 comprende un módulo traductor 32, diseñado para: adquirir, a través de operaciones de interrogación de un tipo “seleccionado”, las selecciones 31 recopiladas por el módulo de gestión 30; convertir dichas selecciones 31 en comandos respectivos 33; y enviar los comandos 33 a la base de datos compartida 25 a través de operaciones de escritura de un tipo “establecido”. El controlador 26 comprende adicionalmente un módulo controlador de estado 34, diseñado para gestionar las transiciones de estado del sistema de aviónica 1 de acuerdo con las selecciones 31 recopiladas por el módulo de gestión 30 y de acuerdo con los cálculos hechos por el navegador 27.

La interfaz de aparato 24 comprende una pluralidad de módulos 35 para gestionar los aparatos a bordo 3, cada uno de los cuales se asocia a un aparato a bordo respectivo 3 y crea una comunicación entre el aparato a bordo 3 en sí mismo y la base de datos compartida 25. En particular, cada módulo de gestión 35 se diseña para adquirir, a través de operaciones de interrogación de un tipo “seleccionado”, los comandos 33 generados por el módulo traductor 32 y dirigido al aparato a bordo respectivo 3, e implementa dichos comandos 33 en el aparato a bordo 3 en sí mismo, manipulando de forma apropiada cualquier posible conflicto entre los comandos 33 y las restricciones operativas del aparato a bordo 3, modificando la ejecución de los comandos 33 de acuerdo con criterios predefinidos.

Dichas modificaciones luego se transfieren en la base de datos compartida 25, sobreescribiendo, a través de operaciones de la escritura del tipo “establecido”, cualesquier posibles parámetros involucrados en estas modificaciones, tales como por ejemplo parámetros operacionales actuales 36 de los aparatos a bordo 3 y parámetros generales 37, tales como parámetros de vuelo y/o aeronave 2, y/o estados operacionales del sistema de aviónica 1.

La base de datos compartida 25 comprende un primer módulo 38 para almacenar los parámetros operacionales actuales 36 y los comandos 33, y un segundo módulo 39 para almacenar los parámetros generales 37. El primer módulo de almacenamiento 38 y el segundo módulo de almacenamiento 39 son interrogados por los módulos de manejo 29 con el propósito de adquirir los parámetros operacionales actuales 36 y los parámetros generales 26 y la visualización de los mismos sobre los dispositivos de interfaz 7.

En uso, el programador 28 activa:

• la interfaz hombre- máquina 23 para adquirir las selecciones 31 hechas por el usuario en los dispositivos de interfaz 7 (100);

• el controlador 26 para adquirir las selecciones 31 desde la interfaz hombre- máquina 23 para convertirlas en comandos respectivos 33 para los aparatos a bordo 3 y escribir los comandos 33 en la base de datos compartida 25 (200);

• la interfaz de aparato 24 para adquirir los comandos 33 desde la base de datos compartida 25 para implementarlos en los aparatos a bordo 3 (300), y para escribir, en la base de datos compartida 25, los parámetros operacionales actuales 36 y los parámetros generales 37, modificados posiblemente luego de la ejecución de los comandos 33 (400); y finalmente

• la interfaz hombre- máquina 23 para adquirir los parámetros operacionales actuales 36 y los parámetros generales 37 desde la base de datos compartida 25 y visualizarlos en los dispositivos de interfaz 7 (500).

Más en detalle, cuando el piloto hace una selección 31 diseñada para un aparato a bordo dado 3 a través del teclado 11 de un dispositivo de interfaz correspondiente 7, el módulo de gestión correspondiente 29 produce un punto de

referencia que representa la selección 31 hecha, que se recopila (101) por el módulo de gestión 30 para hacer la selección 31 que se puede utilizar por el módulo traductor 32, que tiene acceso (201) y convierte (202) dicha selección 31, a través de una validación adecuada que depende del estado operacional actual del sistema de aviónica 1, en un comando correspondiente 33.

El comando 33 de está manera generado se introduce (203) en el primer módulo de almacenamiento 38 para hacerlo utilizable por el módulo 35 para gestionar el aparato a bordo 3 al que se destina el comando 33. Luego, el módulo de gestión 35 tiene acceso (301) al primer módulo de almacenamiento 38, toma el comando 33 y lo implementa en el aparato a bordo correspondiente 3. Luego, en esta forma el módulo 35 que gestiona el aparato a bordo 3 “consume” el punto de referencia inicialmente “producido”, en la forma de selección 31, por el módulo 29 para gestionar el dispositivo de interfaz 7.

En este punto, el módulo 35 para gestionar el aparato a bordo 3 “produce” un parámetro operacional actual 36 que registra el aparato a bordo 3, y, posiblemente, un parámetro general 37, tal como un parámetro de vuelo y/o aeronave 2, y/o un estado operacional del sistema de aviónica 1.

Los parámetros operacionales actuales 36 y los parámetros generales 37 luego se ingresan (401, 402), respectivamente, en el primer módulo de almacenamiento 38 y en el segundo módulo de almacenamiento 39, a fin de hacer dichos parámetros operacionales actuales 36 y parámetros generales 37 utilizables por aquellos módulos 29 para gestionar los dispositivos de interfaz 7 que están involucrados en el uso de los parámetros operacionales actuales 36 y parámetros generales 37 en sí mismos. Luego, los módulos de manejo 29 tienen acceso (501, 502), sin la intermediación del módulo traductor 32, los parámetros operacionales actuales 36 y los parámetros generales 37, y los visualizan en los dispositivos de interfaz respectivos 7. De esta forma, los módulos 29 para gestionar los dispositivos de interfaz 7 se convierten en “consumidores” de los datos “producidos”, en la forma de parámetros operacionales actuales 36 y de parámetros generales 37, por el módulo 35 para gestionar el aparato a bordo 3.

Como se puede observar, se transmite la secuencia lógica seguida por el programador 28, por medio de relaciones de uso entre los diversos módulos, un punto de referencia entre la interfaz hombre- máquina 23 y la interfaz de aparato 24 a lo largo de dos rutas distintas dependiendo de si el punto de referencia producido se constituye por una selección 31, o bien por un parámetro operacional actual 36 o un parámetro general 37.

A partir de la descripción anterior, es evidente una función de desacoplamiento desarrollada por el módulo de gestión 30, por el módulo traductor 32, y por la base de datos compartida 25. En particular, la base de datos compartida 25 y el módulo de gestión 30 funcionan como objetos pasivos en la medida en que están disponibles, por medio de operaciones de “seleccionar” y “establecer, las selecciones 31 y los comandos 33 al módulo traductor 32, que en cambio funciona como un objeto activo. Por lo tanto, dicho desacoplamiento permite evitar las relaciones directas de uso para el intercambio de un punto de referencia, constituido por una selección 31, un comando 33, un parámetro operacional actual 36, o bien un parámetro general 37, entre un módulo 29 para gestionar un dispositivo de interfaz 7, que inicialmente es un dispositivo para producir datos y luego se convierte en un consumidor de datos, y un módulo 35 para gestionar un aparato a bordo 3, que inicialmente es un consumidor de datos y luego se convierte en un dispositivo para producir datos.

Esto conduce a la ventaja evidente de que la adición de un aparato a bordo 3 o de un dispositivo de interfaz 7 simplemente requiere la adición de un módulo de gestión 35 o 29 y posiblemente la actualización de la cartografía entre selecciones 31 y comandos 33 en el módulo traductor 32 en la medida en que no se ha modificado las relaciones de uso entre la interfaz hombre- máquina 23, la base de datos compartida 25, y el controlador 26.

Adicionalmente, la secuencia lógica seguida por el programador 28 se comparte en un número finito de procesos concurrentes con prioridades y frecuencias definidas. En particular, el módulo traductor 32 realiza sus tareas dentro de un proceso de prioridad y frecuencia máxima, para garantizar la secuencia lógica correcta.

Por último cabe notar que la construcción del programa de operación de vuelo 22 en el lenguaje de programación Ada 95 permite explotación ventajosa de las características intrínsecas de este lenguaje de programación, es decir acceso atómico a los datos (selección 31, comando 33, parámetro operacional actual 36, o parámetro general 37) y protección desde el acceso actual en lectura y escritura mediante procesos paralelos, por ejemplo mediante los módulos 29, 32, 35, que se crean como desacoplamiento de refuerzo de “objetos protegidos”, entre la interfaz de hombre- máquina 23 y la interfaz de aparato 24.

En base a lo que se ha descrito anteriormente, es por lo tanto posible establecer inmediatamente que la presente invención logra una arquitectura de software que permite que sean alcanzados todos los objetivos de diseños indicados anteriormente, a saber:

• capacidad de implementar una pluralidad de dispositivos de interfaz de hombre-máquina y una variedad de sensores/accionadores a través de una abertura y configuración modular, donde el número de dispositivos de

interfaz de hombre-máquina y el número de sensores/accionadores son funcciones del nivel de seguridad, y por lo tanto del nivel de redundancia, solicitado por la plataforma bajo desarrollo,

• comunicación entre una pluralidad de dispositivos de interfaz de hombre-máquina y sensores/accionadores que se alcanza a través de una pluralidad de instancias de una misma clase de software únicamente definido,

• desacoplamiento de la arquitectura de software entre la pluralidad de dispositivos de interfaz de hombre-máquina y la pluralidad de sensores/accionadores que permite alta capacidad de mantenimiento de la aplicación de software y facilidad de expansión,

• capacidad de resolver posibles conflictos de acceso a las estructuras de datos compartidos, que se pueden alcanzar a través de una matriz de regla/prioridad, y

• una aplicación de software de conformidad con los requerimientos de proceso necesarios para apoyar los procedimientos de certificación asociados con aplicación de software crítico de seguridad, de acuerdo con el estándar RTCA-DO178B en el caso específico de una aeronave.

REIVINDICACIONES

1. Un sistema de seguridad crítico muy dinámico (1) que comprende por lo menos un aparato electrónico (3), por lo menos un dispositivo de interfaz hombre-máquina (7) diseñado para permitir que un usuario interactúe con por lo menos un aparato electrónico (3); y un dispositivo de procesamiento (5) conectado a por lo menos un aparato electrónico (3) y por lo menos un dispositivo de interfaz hombre-máquina (7) y programado para gestionar intercambio de datos en el sistema de seguridad crítico muy dinámico (1); en donde el dispositivo de procesamiento (5) se diseña para almacenar y ejecutar un programa de software de gestión de intercambio de datos con base en módulos que incluye los siguientes módulos de software:

• una interfaz hombre- máquina (23) diseñada para visualizar el dispositivo de interfaz hombre-máquina (7);

• una interfaz de aparato (24) diseñada para visualizar el aparato electrónico (3);

• una base de datos compartida (25) diseñada para almacenar los datos que se van a compartir entre la interfaz hombre- máquina (23) y la interfaz de aparato (24);

• un controlador (26) diseñado para gestionar el intercambio de los datos que se van a compartir entre la interfaz hombre- máquina (23) y la interfaz de aparato (24); y

• un programador (28) diseñado para programar la operación de la interfaz hombre- máquina (23), la interfaz de aparato (24), la base de datos compartida (25) y el controlador (26);

en donde la interfaz hombre- máquina (23) comprende los siguientes módulos:

• un administrador de interfaz (29) para cada dispositivo de interfaz hombre-máquina (7) y diseñado para establecer una comunicación entre el dispositivo de interfaz hombre-máquina respectivo (7) y la base de datos compartida (25); y

• un administrador de selección (30) diseñado para gestionar selecciones hechas por un usuario;

en donde cada administrador de interfaz (29) se diseña para:

- adquirir una entrada de datos (31) en el dispositivo de interfaz correspondiente hombre-máquina (7);

- enviar la entrada de datos (31) al administrador de selección (30);

- adquirir los datos (36, 37) almacenados en la base de datos compartida (25) para el dispositivo de interfaz hombre-máquina respectivo (7); e

- ingresar los datos adquiridos en el dispositivo de interfaz hombre-máquina respectivo (7);

y en donde el administrador de selección (30) se diseña para:

- recopilar las entradas de datos (31) hechas en el dispositivo de interfaz hombre- máquina (7); y

- enviar las entradas de datos recopilados al controlador (26);

en donde el controlador (26) comprende el siguiente módulo:

• un traductor (32) diseñado para:

- adquirir una entrada de datos (31) enviada por el administrador de selección (30);

- validar la entrada de datos adquirida (31) de acuerdo con un estado operacional actual del sistema de seguridad crítico muy dinámico (1);

- transformar la entrada de datos adquirida (31) en un comando (33) para el aparato electrónico (7); y

- almacenar el comando (33) en la base de datos compartida (25) para que esté disponible para la interfaz de aparato (24);

y en donde la interfaz de aparato (24) comprende el siguiente módulo:

• un administrador de aparato (35) para cada aparato electrónico (3) y diseñado para establecer una comunicación entre el aparato electrónico respectivo (3) y la base de datos compartida (25); en donde cada administrador de aparato (35) se diseña para:

- adquirir un comando (33) almacenado en la base de datos compartida (25) para el aparato electrónico respectivo (3);

- legalizar el comando adquirido (33) en el aparato electrónico respectivo (3); y

- generar y almacenar en la base de datos compartida (25) los datos que se relacionan con la operación del aparato electrónico respectivo (3) y/o del sistema de seguridad crítico muy dinámico (1) para que esté disponible a la interfaz hombre- máquina (23);

por lo cual la gestión de intercambio de programa de software de datos se convierte en configurable de tal manera que una adición de un aparato electrónico (3) o del dispositivo de interfaz hombre-máquina (7) requiere una adición de un administrador de aparato respectivo (35) o de un administrador de interfaz (29) y posiblemente la actualización de cartografía entre las entradas de datos (31) y comandos (33) en el traductor (32), en la medida en que no se tengan que modificar las relaciones de uso entre la interfaz hombre- máquina (23), la base de datos compartida (25), y el controlador (26).

2. El sistema de la reivindicación 1, en donde el administrador de selección (35) se diseña adicionalmente para:

- resolver los accesos concurrentes en conflicto con el aparato electrónico (3).

3. El sistema de las reivindicaciones 1 o 2, en donde cada administrador de aparato (35) se diseña adicionalmente para:

- gestionar apropiadamente los conflictos entre comandos (33) y restricciones operativas del aparato electrónico (3); y

- en caso de conflictos entre comandos (33) y restricciones operativas del aparato electrónico (3), modificar la ejecución de los comandos (33) de acuerdo con lo anterior en base a criterios predefinidos, y modificar los datos (36, 37) almacenados en la base de datos compartida (25) y afectados por la modificación de la ejecución de los comandos (33).

4. El sistema de cualquier reivindicación precedente, en donde la base de datos compartida (25) comprende los siguientes módulos:

• un almacenamiento de aparato (38) diseñado para almacenar comandos (33) y datos (36) que se relacionan con el aparato electrónico (3); y

• un almacenamiento de sistema inicuamente de (39) para almacenar datos (37) que se relaciona con el sistema de seguridad crítico muy dinámico (1).

5. El sistema de cualquier reivindicación precedente, en donde el programador (28) se diseña para operar de forma secuencial:

• la interfaz hombre- máquina (23) para adquirir una entrada de datos (31) hecha en un dispositivo de interfaz hombre-máquina (7);

• el controlador (26) para adquirir y transformar la entrada de datos (31) en un comando (33) para un aparato electrónico (3) y enviar el comando (33) a la base de datos compartida (25);

• la base de datos compartida (25) para almacenar el comando (33) para hacerlo disponible a la interfaz de aparato (24);

• la interfaz de aparato (24) para adquirir el comando (33) desde la base de datos compartida (25) y ejecutarlo en el aparato electrónico (3); y

• la interfaz hombre- máquina (23) para adquirir parámetros (36, 37) almacenados en la base de datos compartida (25) y visualizarlos en el dispositivo de interfaz hombre-máquina (7).

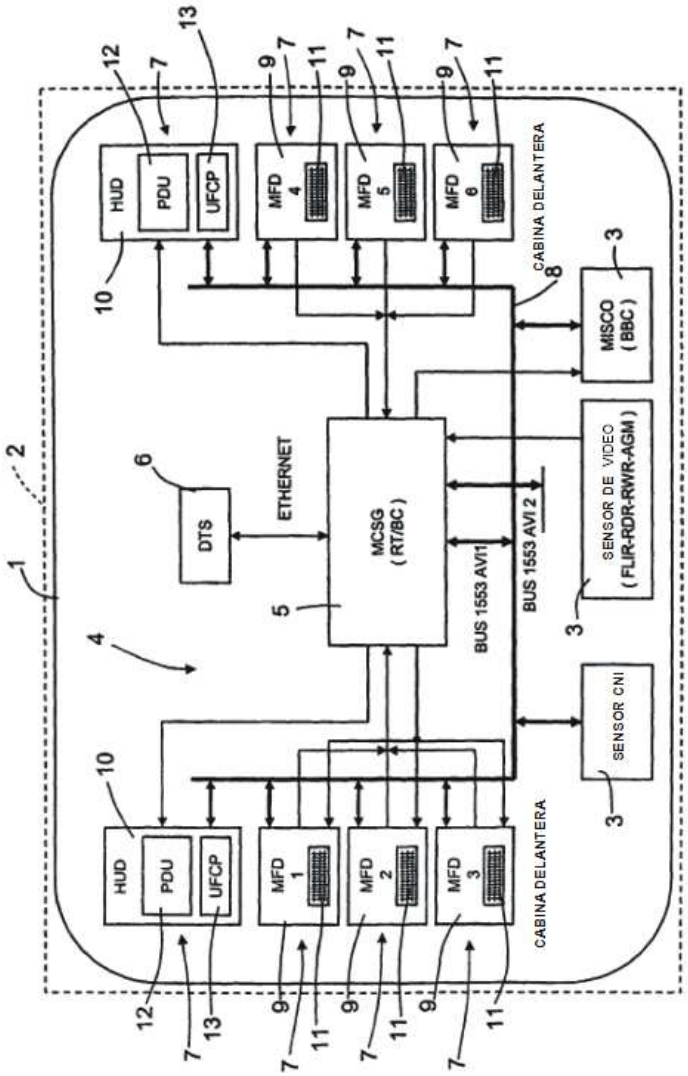
HUD
PDU
UFCP
MFD 4
MFD 5
MFD 6
CABINA DELANTERA
MISCO (BBC)
SENSOR DE VIDEO (FLIR-RDR-RWR-AGM)
SENSOR CNI
DTS
ETHERNET
MCSG (RT/BC)
BUS 1553 AVI1
BUS 1553 AVI2
MFD 1
MFD 2
MFD 3
CABINA DELANTERA

Fig.1

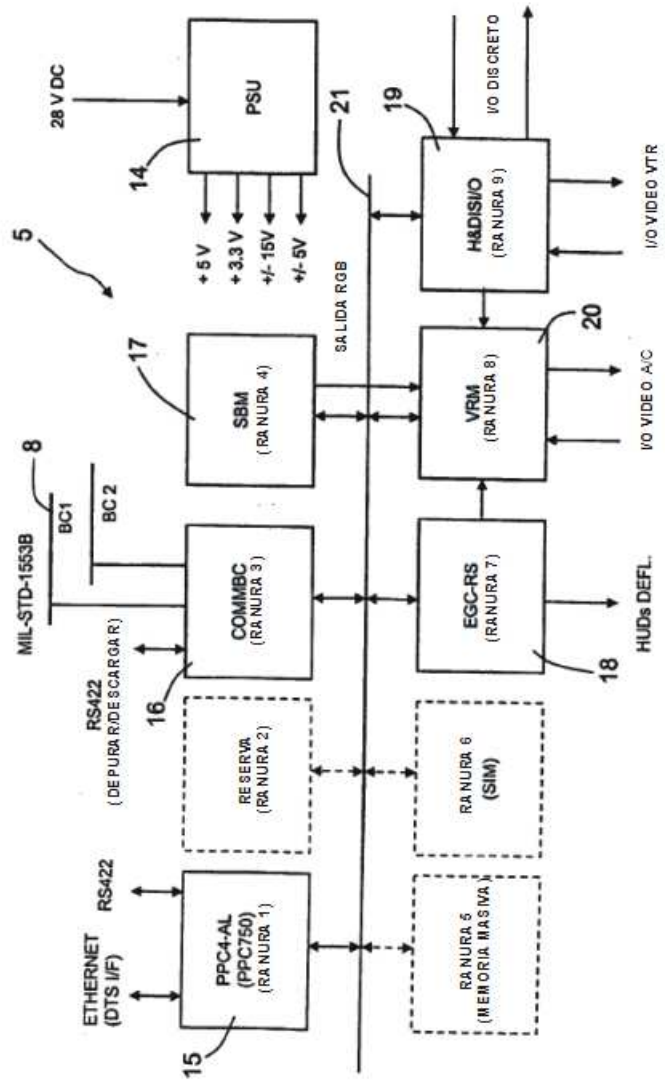
28 V DC
PSU
14
+ 5 V
+ 3.3 V
+/- 15V
+/- 5V
5
21
19
I/O DISCRETO
H&DISI/O
(RA NURA 9)
I/O VIDEO VTR
SALIDA RGB
17
SBM
(RA NURA 4)
VRM
(RA NURA 8)
20
I/O VIDEO A/C
8
BC1
BC 2
MIL-STD-1553B
COMMBC
(RA NURA 3)
EGC-RS
(RA NURA 7)
HUDs DEFL.
18
RS422
(DEPURA R/DESCARGA R)
16
RESERVA
(RA NURA 2)
RA NURA 6
(SIM)
RS422
ETHERNET
(DTS I/F)
PPC4-AL
(PPC750)
(RA NURA 1)
RA NURA 5
(MEMORIA MASIVA)
15

Fig.2

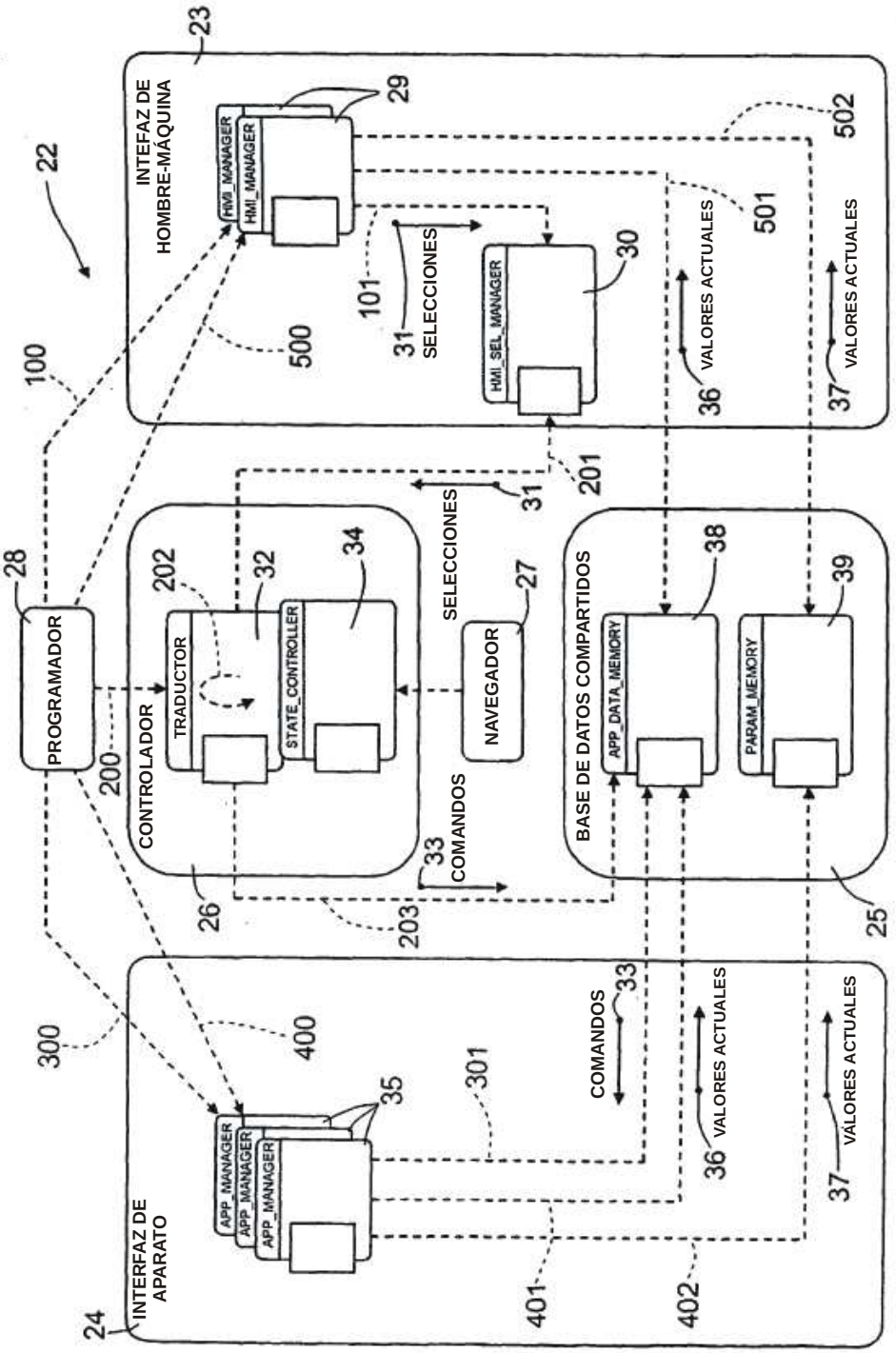
22
23
INTEFAZ DE HOMBRE-MÁQUINA
HMI_MANAGER
HMI_MANAGER
29
502
501
500
101
31
SELECCIONES
HMI_SEL_MANAGER
30
36
VALORES ACTUALES
37
VALORES ACTUALES
100
201
31
SELECCIONES
28
PROGRAMADOR
202
32
34
TRADUCTOR
STATE_CONTROLLER
CONTROLADOR
200
NAVEGADOR
27
38
39
BASE DE DATOS COMPARTIDOS
APP_DATA_MEMORY
PARAM_MEMORY
33
COMANDOS
26
203
25
300
400
35
301
APP_MANAGER
APP_MANAGER
APP_MANAGER
INTERFAZ DE APARATO
24
COMANDOS
33
36
VALORES ACTUALES
37
VALORES ACTUALES
401
402

Fig.3