



(19)



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA

(11) Número de publicación: **2 307 954**

(51) Int. Cl.:
B61L 21/00 (2006.01)
B61L 27/00 (2006.01)

(12)

TRADUCCIÓN DE PATENTE EUROPEA

T3

(96) Número de solicitud europea: **03742528 .7**
(96) Fecha de presentación : **18.02.2003**
(97) Número de publicación de la solicitud: **1490253**
(97) Fecha de publicación de la solicitud: **29.12.2004**

(54) Título: **Método y dispositivo para generar unidades de control lógicas para aparatos informáticos esenciales con base en estaciones ferroviarias.**

(30) Prioridad: **22.02.2002 IT SV02A0009**

(73) Titular/es: **ALSTOM FERROVIARIA S.p.A.**
Via Ottaviano Moreno, 23
12038 Savigliano, Cuneo, IT

(45) Fecha de publicación de la mención BOPI:
01.12.2008

(72) Inventor/es: **Minkowitz, Cydney y**
Tramontana, Francesco

(45) Fecha de la publicación del folleto de la patente:
01.12.2008

(74) Agente: **Fernández Lerroux, Aurelio**

Aviso: En el plazo de nueve meses a contar desde la fecha de publicación en el Boletín europeo de patentes, de la mención de concesión de la patente europea, cualquier persona podrá oponerse ante la Oficina Europea de Patentes a la patente concedida. La oposición deberá formularse por escrito y estar motivada; sólo se considerará como formulada una vez que se haya realizado el pago de la tasa de oposición (art. 99.1 del Convenio sobre concesión de Patentes Europeas).

DESCRIPCIÓN

Método y dispositivo para generar unidades de control lógicas para aparatos informáticos esenciales con base en estaciones ferroviarias.

La invención se refiere a un método para generar unidades de control lógicas para aparatos informáticos esenciales con base en estaciones ferroviarias, es decir, en unidades de control de sistema de estaciones ferroviarias, tal como se describe en el preámbulo de la reivindicación 1, y a un aparato estacionario informático esencial para llevar a cabo dicho método.

Un método y un aparato estacionario informático esencial de este tipo se conoce a partir de la solicitud de patente italiana número ITGE94A000061.

Los sistemas de estaciones incluyen generalmente una unidad central que genera controles para diferentes elementos de maniobra, tales como señales y/o agujas y/o circuitos de vía, o similares. Con el fin de garantizar que el tren pueda desplazarse de manera segura, estos elementos de maniobra deben adoptar diferentes estados, tales como "vía abierta" o una señal de stop o el cambio de aguja de una vía férrea, según una cierta lógica, que tiene en cuenta los estados o el cambio de estado de otros elementos de maniobra que, cuando se llevan a o se dejan en ciertos estados, pueden provocar colisiones o situaciones peligrosas, o incluso simplemente no cumplir con las regulaciones de operaciones de circulación de trenes admitidas.

Los elementos de maniobra están dotados generalmente de accionadores que realizan operaciones de cambio de estado y de dispositivos de control y/o monitorización y/o de diagnóstico que envían señales acerca del estado actual y la funcionalidad del mismo, de modo que el aparato estacionario con base en la estación, es decir, la unidad de control central puede tener todos los ajustes del tráfico de la vía férrea bajo control todo el tiempo. Por tanto, un control de cambio de estado predeterminado transmitido a un elemento particular genera una cadena de controles de cambio o mantenimiento de estado para otros elementos de maniobra según reglas bien definidas. Por consiguiente, la unidad de control central no sólo tiene subunidades de salida de controles para comunicarse con cada uno de los diferentes elementos de una manera dedicada, tanto para enviar controles como para recibir una retroalimentación, sino que también debe funcionar bajo una lógica estricta, que incorpora reglas de cambio de estado de elementos de maniobra, conforme a operaciones de garantía de seguridad. Estas redes pueden realizarse, y de hecho se han realizado, meramente mediante hardware, es decir, redes de circuitos conectados a una pluralidad de componentes de hardware diseñados para realizar operaciones booleanas predeterminadas. Normalmente, en aplicaciones de vía férrea, los componentes diseñados para realizar operaciones booleanas consistían en circuitos de conexión de relé o circuitos integrados lógicos conectados y diseñados especialmente para generar salidas de controles compatibles con reglas de cambio de estado de elementos de maniobra.

Cuando se introdujeron los ordenadores en aplicaciones de vía férrea, las unidades lógicas de hardware se sustituyeron progresivamente por programas de monitorización y control que incluyen conjuntos de ecuaciones booleanas, que describen los comportamientos de los operadores booleanos de hardware individuales y forman, cuando están integrados de manera apropiada en un programa de control lógico, una unidad lógica virtual equivalente a hardware.

Un ordenador esencial central puede incluir diferentes procedimientos de biblioteca normalizados, por ejemplo controladores para generar controles de cambio de estado, programas para gestionar funciones de diagnóstico, control y monitorización que incorporan estructuras de control y monitorización y reproducen reglas generales de movimiento regulado seguro. Sin embargo, estos programas de gestión general requieren una adaptación especial basándose en la estructura particular del sistema de la estación, es decir, de sus elementos de maniobra, y en reglas de cambio de estado relacionadas, las denominadas tablas de estado. Para ello, cada unidad central requiere un programa de control lógico para relacionar las operaciones de control y monitorización con las condiciones del entorno, tal como se define por la estructura del sistema de la estación. Estas lógicas de control no pueden prefabricarse sino que dependen de la aplicación, es decir, dependen de la construcción del sistema de estación específico.

Se sabe que las lógicas de control, compuestas por conjuntos de ecuaciones booleanas, cuyas variables vienen dadas por los estados de los diferentes elementos y por los controles de estado y datos de diagnóstico de los mismos, se generan por sistemas automáticos, es decir, programas de generación que generan los conjuntos de ecuaciones booleanas que forman los algoritmos de programas lógicos de monitorización y control específicos de la estación utilizando tablas de estado o tablas de cambio de estado y el diagrama del sistema de estación como base de entrada o conocimiento.

En la técnica anterior, el método proporciona la implementación de la lógica de control así obtenida en el ordenador esencial del módulo de monitorización y control lógico, y una comprobación funcional consiguiente, posiblemente editando el programa lógico cuando se producen errores o incompatibilidades de estado entre elementos de maniobra. Esta comprobación funcional incluye normalmente ensayos *in situ*, es decir se realiza cuando la unidad de monitorización y control está instalada en el sistema de estación específico.

El modo de comprobación es relativamente complejo y lleva mucho tiempo. Además, cuando la unidad lógica no es un producto de software, sino que está compuesta por un conjunto de componentes electrónicos diseñados para realizar

ES 2 307 954 T3

funciones booleanas, la implementación es aún más difícil, porque el circuito debe construirse antes de comprobar el funcionamiento del mismo.

5 La invención tiene el objetivo de mejorar un método tal como se ha descrito anteriormente para reducir los requisitos de tiempo y simplificar las operaciones de comprobación, mientras se mantiene una alta seguridad de funcionamiento de la unidad lógica, conforme a reglas de cambio de estado de elementos de maniobra y a la estructura de la estación.

10 La invención consigue los propósitos anteriores proporcionando un método tal como se ha descrito anteriormente en el presente documento, que incluye las etapas siguientes:

15 generación paralela de dos unidades de control lógicas, según el mismo diagrama de estación y la misma tabla de estado, generándose las dos unidades mediante dos programas de generación que son lo más diferentes posible entre sí;

comparación entre las redes de circuitos lógicos o los programas lógicos simuladores de red proporcionados por los dos programas de generación diferentes para comprobar diferencias entre los mismos.

20 Si no se encuentra ninguna diferencia, las ecuaciones booleanas de la unidad de monitorización y control lógica deberían considerarse correctas. Si se detectan diferencias, deberían hacerse cambios y correcciones.

Estos cambios y correcciones pueden consistir incluso en comprobar que las tablas de estado y el diagrama de estación están codificados apropiadamente en un formato comprensible para los programas de generación.

25 Los dos programas de generación lógicos son independientes, y pueden diferir tanto en términos de lenguajes de programación como en términos de análisis sistemático de variables y enfoque de lectura.

30 En sistemas de estaciones muy grandes se proporciona un gran número de variables y, aunque se pretende que el algoritmo sólo realice operaciones simples, el número de elementos de maniobra y las conexiones lógicas entre estados pueden requerir una tarea de procesamiento difícil. En este caso, pueden usarse algoritmos de procesamiento que consisten en denominadas redes neuronales, a las que se proporcionan como base de conocimiento listas de elementos de maniobra y tablas de estado que gobiernan las relaciones de estado o de cambio de estado entre elementos de maniobra. Las redes neuronales proporcionan la ventaja considerable de que extienden su conocimiento uso tras uso, ya que la base de conocimiento y la interpretación de la misma aumenta progresivamente, y como resultado de
35 ello se cambian los modos de cálculo. Además, las redes neuronales usan la base de conocimiento sustancialmente independientemente de la estructura específica de la tabla de estado y el diagrama de estación, y generalmente pueden reconocer situaciones idénticas o similares y usarlas como experiencia para manejar situaciones nuevas que presenten analogías con situaciones de la base de conocimiento.

40 Las unidades de control lógicas generadas por los dos programas de generación, cuando se proporcionan en forma de software, consisten en un conjunto de ecuaciones cuya generación se basó en la tabla de estado y en información relacionada con elementos de la estación.

45 La información relacionada con elementos de la estación incluye el tipo de entradas y salidas requeridas por los elementos de la estación, un código de ID y un programa de control, es decir, un controlador para convertir el control generado por la unidad lógica en un control comprensible para el elemento de maniobra y las etiquetas de tiempo.

50 Debería indicarse que, como en la técnica anterior, la lógica de control es independiente del tipo de controlador específico, y que sólo tiene que conocer variables de entrada de control y/o de salida de control.

55 Esto permite usar el método también en unidades de control existentes cuando ha de ampliarse la estación. En este caso, se genera una nueva lógica de control para tener en cuenta las variaciones, y no tiene que proporcionarse ningún controlador u otros componentes aparte de los que existen en las memorias de la unidad de control central, en secciones dedicadas especialmente, que la unidad de control lógica llama de nuevo o encamina de forma apropiada, cuando ha de manejarse el elemento relevante.

60 En este caso, los programas de generación, así como los módulos de lectura o introducción de diagrama de estación y tabla de estado pueden formar una sección estable en el software de gestión de la unidad de control central, es decir, el aparato estacionario informático esencial.

Como programa para comparar las unidades lógicas resultantes, es decir, las ecuaciones de algoritmos que definen unidades lógicas, pueden usarse productos de software de comparación, por ejemplo MKS Visual Difference para Win 32 - Versión 3.2b de Mortice Kern Systems Inc. y/o Microsoft® WinDiff - Versión 4.0 de Microsoft Corp.

65 Si se obtiene un resultado coincidente a partir de la comparación entre las unidades de control lógicas generadas por los dos programas de generación diferentes, es decir si no se detecta ninguna diferencia, se considera que la unidad de control lógica generada es correcta.

ES 2 307 954 T3

Si las dos unidades no coinciden, el programa de comparación proporciona una lista de diferencias que deberán analizarse para hacer correcciones, tras lo cual las etapas de generar la unidad de control lógica por los dos programas de generación diferentes y de compararlas debería repetirse.

La ventaja de este modo de comprobación es considerable, puesto que puede realizarse sin la necesidad de implementar realmente la lógica de control en el sistema, realizándose las operaciones de comprobación directamente y sólo en los datos virtuales del ordenador usado para generar la unidad de control lógica. Ni siquiera es necesario cargar la unidad de control lógica generada en la unidad de control central, y tampoco es necesario establecer una interfaz de la misma con los programas de control y los controladores contenidos en los mismos.

Según otra característica, los programas de generación de lógica de control reciben una entrada que no sólo incluye datos de variables relacionadas con el estado de los elementos de maniobra, sino también datos de variables relacionadas con el estado de las señales de monitorización, que se proporcionan por elementos de maniobra como una salida a la unidad de control central.

Además de los datos relacionados con las variables de monitorización, los programas de generación de la lógica de control también reciben datos relacionados con las variables de diagnóstico de los elementos de maniobra.

Según una mejora adicional, el programa de comparación y/o el segundo programa de generación y/o ambos programas de generación incluyen rutinas para visualizar los errores encontrados, que se proporcionan como mensajes de error.

En este caso, de nuevo, pueden proporcionarse rutinas de corrección, para iniciarse por el usuario a voluntad o saltárselas si el usuario decide realizar correcciones estructuradas y orgánicas al final del procedimiento de generación y/o comparación.

Según otra mejora más, puesto que los dos programas de generación de lógica de control deben ser al menos ligeramente diferentes, al menos uno de los dos programas de generación puede incluir una rutina de inicio para analizar los datos de entrada, es decir, la tabla de estado y/o la tabla de cambio de estado y/o la lista de elementos de maniobra en el diagrama del sistema de estación específico.

En este caso, se comprueba la consistencia estructural de los datos de entrada anteriores tanto respecto a la codificación como a la estructura de los mismos, y con respecto a la presencia de errores o contradicciones lógicas, tales como claves que identifiquen elementos de maniobra no únicos, combinaciones prohibidas o imposibles de elementos de maniobra que se requieren por el sistema de estación, etc. Por tanto, en esta fase preliminar, se garantiza una consistencia perfecta para la base de datos de entrada que forma la base de conocimiento de los programas de generación de unidad de control lógica.

Debe indicarse que el método inventivo permite integraciones muy sencillas en sistemas de estaciones en las que deben añadirse elementos de maniobra. De hecho, puesto que se proporcionan unidades de controlador prefabricadas para cada elemento de maniobra, pueden añadirse nuevos elementos de maniobra de manera sencilla actualizando el diagrama del sistema de estación, es decir, la lista de elementos y las tablas de estado, y generando en paralelo dos unidades de control lógicas, y comparándolas para actualizar la unidad de control central con la nueva situación del sistema de estación.

Los programas que generan la unidad de control lógica sustancialmente no están relacionados con los tipos de elemento de maniobra, y no requieren conocer los controladores de elemento de maniobra individuales, ni los sistemas de diagnóstico y monitorización, sino que sólo requieren la indicación del número y tipo de los datos de control que deben proporcionarse al elemento de maniobra y los datos de diagnóstico y de monitorización que deben transmitirse por el elemento de maniobra o las unidades de controlador del mismo. La conformidad de esas variables de diagnóstico o de monitorización y control con el elemento de maniobra se garantiza por el controlador específico que convierte las variables de diagnóstico y de monitorización y control en la estructura requerida por el hardware de elemento de maniobra - en lo que se refiere a las variables de control - y en la estructura requerida por la unidad de control central - en lo que se refiere a las variables de diagnóstico y de monitorización.

La invención también se refiere a un aparato de control esencial de estación de vía férrea (un denominado ASCV), que está diseñado para formar la unidad de control central para una pluralidad de elementos de maniobra de un sistema de estación, aparato informático esencial con base en la estación de vía férrea que incluye entradas para señales de diagnóstico y de monitorización generadas por los elementos de maniobra, salidas para señales de control de cambio de los elementos de maniobra, un programa de control que tiene un controlador para cada tipo diferente de elemento de maniobra, es decir, un programa para controlar y establecer una interfaz para las variables de control generadas por el aparato de estación y transmitidas a los elementos de maniobra y/o las variables de diagnóstico generales y/o de monitorización generadas por los elementos de maniobra y transmitidas al aparato de estación, un diagrama del sistema de estación, es decir, una base de conocimiento que contiene una lista de los elementos de maniobra del sistema de estación y las relaciones entre los mismos, una base de datos de reglas de cambio de estado o adopción de estado admitidas para los diferentes elementos de maniobra según requisitos de gestión del tráfico de la vía férrea de seguridad, la denominada tabla de estado, una unidad de control lógica que incluye algoritmos que consisten en

ES 2 307 954 T3

ecuaciones booleanas y/o funciones lógicas para una concatenación y transmisión de controles adecuada de secuencias de controles de elementos de maniobra según el diagrama del sistema de estación y la tabla de estado.

5 Según la invención, el aparato informático esencial con base en la estación incluye además un programa para la generación automatizada y redundante de los algoritmos que forman la unidad de control lógica, que usa redundancia para realizar una comprobación de software de dichos algoritmos de la unidad de control lógica.

10 Este programa forma una rutina el usuario que puede llamar de nuevo siempre que se cambie el diagrama del sistema de estación, es decir, cuando se añadan o eliminen elementos de maniobra y/o se cambien reglas de gestión del tráfico de la estación, es decir, la tabla de estado.

15 Se genera redundancia usando dos programas diferentes para generar los algoritmos booleanos que forman la unidad de control lógica, programas que proporcionan dos unidades de control lógicas, cuyos algoritmos, es decir, ecuaciones booleanas, se comparan, y se consideran correctas cuando de la comparación no resulta ninguna diferencia entre los algoritmos de generación.

20 La corrección de los algoritmos de la unidad de control lógica se garantiza totalmente proporcionando dos programas de generación que son diferentes en cierta medida, proporcionándose su nivel de diferencia usando dos lenguajes de programación diferentes para los programas de generación y/o desarrollando los dos programas de generación por dos equipos de desarrollo diferentes y/o usando diferentes estructuras de datos de entrada, es decir, datos de tabla de estado y/o diagrama del sistema de estación, que no obstante concuerdan con restricciones de la tabla de estado y el diagrama del sistema de estación, siendo estas últimas las mismas para ambos programas de generación.

25 Mejoras de la invención formarán el contenido de las reivindicaciones dependientes.

Las ventajas de la invención resultarán más evidentes a partir de la siguiente descripción de una realización no limitativa que se describe basándose en las figuras adjuntas, en las que:

30 La figura 1 es un diagrama de flujo del método de la invención.

La figura 2 es un diagrama de bloques de un sistema de estación, que tiene un aparato informático esencial según esta invención.

35 Con referencia a la figura 1, el método de la invención proporciona la generación automatizada y redundante de la lógica de monitorización y control del sistema de estación, es decir una unidad central para controlar y supervisar los diferentes elementos, tales como luces, aguas de vía férrea, circuitos de vía, o similares, ubicados en una estación particular. La unidad de monitorización y control central, que se denomina aparato estacionario informático esencial con base en la estación incluye generalmente dos niveles de monitorización y control lógicos. Las lógicas generales de control, monitorización y posiblemente diagnóstico orientadas al procedimiento consisten en programas orientados al procedimiento que son independientes de sistemas de estación específicos y de la estructura de los mismos, así como del número y los tipos de elementos y/o de los requisitos particulares del tráfico de la vía férrea. Normalmente, estos programas usan estructuras lógicas que transmiten datos de salida booleanos y reciben datos de entrada booleanos que tienen significados de verdadero/falso.

45 Estos programas orientados al procedimiento universales no pueden funcionar de manera apropiada en todos los sistemas, y requieren el procesamiento de datos lógicos, particularmente controles y retroalimentaciones, así como datos de diagnóstico, que están estructurados conforme a las configuraciones específicas del sistema de estación de vía férrea. Además, cualquier sistema de estación específico debe realizar operaciones de gestión del tráfico de la vía férrea específicas, que deben llevarse a cabo según reglas de gestión de seguridad predeterminadas. Estas reglas requieren la concatenación de controles de adopción de estado hacia, y la realización apropiada de retroalimentaciones desde, los diferentes elementos según predeterminados diagramas que no sólo dependen del sistema específico, sino también de normas de cambio y control de estado, y de reglas de ejecución de movimientos, que varían en función del caso y dependiendo de las organizaciones de gestión del tráfico de la vía férrea.

55 Con referencia a lo anterior y a la figura 1, la invención se refiere a un método para la generación automatizada de dichas lógicas de monitorización y control específicas del sistema de estación, que incluye una primera etapa para procesar datos de diagrama del sistema de estación y reglas de cambio de estado y/o gestión de control de elementos, de una forma comprensible, desde un programa para generar dicha lógica de control y monitorización y/o diagnóstico. Por tanto, se generan dos bases de datos, una para la configuración del sistema y la otra para reglas de cambio y/o adopción de estado de elementos que tiene en cuenta las relaciones o concatenaciones de controles con otros elementos que posiblemente se ven afectados por el control de un primer elemento.

65 La base de datos de configuración de construcción del sistema de estación y la base de datos de tabla de estado (reglas de cambio de estado o adopción de estado para los diferentes elementos) forman la denominada base de conocimiento para un algoritmo para generar la lógica de control y monitorización y/o diagnóstico para el sistema de vía férrea específico.

ES 2 307 954 T3

A continuación, se transmiten los datos como una base de datos de conocimiento a un programa para analizar y generar ecuaciones booleanas de control y monitorización, ecuaciones que sustancialmente constituyen los algoritmos que forman el programa de control y monitorización. Paralelamente a esto, se transmiten los mismos datos de base de conocimiento relativos al sistema de estación y a la tabla de estado a un segundo programa de análisis y procesamiento que genera un segundo conjunto de ecuaciones booleanas, para formar un segundo programa lógico de control y monitorización relativo a la misma estación y basándose en las mismas reglas de gestión que el primer programa.

A continuación, se comparan los dos conjuntos de ecuaciones booleanas mediante algoritmos de comparación. El resultado de comparación determina si las ecuaciones booleanas que forman el núcleo del programa lógico de control y monitorización específico de la estación se han generado correctamente o si se han producido errores de generación.

Cuando se descubre que los dos conjuntos de ecuaciones booleanas son idénticos, se consideran correctas, y se considera que el programa lógico de control y monitorización tiene una seguridad certificada. Cuando se detectan diferencias, el programa de comparación transmite mensajes de informe de diferencias, que también pueden incluir notas de mensajes de error o indicaciones específicas de las diferencias detectadas y de los errores que posible o probablemente han provocado las diferencias.

En el último caso, se requiere una acción de corrección acción, tras la que debe repetirse el proceso de generación.

La etapa de generación y comparación redundante sustituye de manera segura las etapas de comprobación de la técnica anterior, que se llevan a cabo cuando el programa lógico de control y monitorización se carga en la unidad de control central y cuando las comprobaciones *in situ* funcionales se llevan a cabo directamente en el sistema de estación, implicando así desventajas de costes y duración.

La etapa de comprobación basada en la generación y comparación redundante de conjuntos redundantes de ecuaciones booleanas se realiza o bien en el mismo ordenador que el ordenador de generación o bien en un ordenador dedicado, y es relativamente rápida. La generación paralela puede realizarse en paralelo en el tiempo o bien en el mismo ordenador o bien en ordenadores separados.

Las diferencias entre los programas para generar conjuntos de ecuaciones booleanas deben establecerse a diferentes niveles. Esto puede obtenerse usando diferentes lenguajes de programación o desarrollando dichos programas de generación por equipos diferentes de desarrolladores. Por ejemplo, cuando se usan redes neuronales, existe un gran número de redes, proporcionadas por diferentes desarrolladores, que analizan bases de conocimientos según diferentes reglas, y que generalmente proporcionan resultados idénticos, aunque en momentos ligeramente diferentes.

Obviamente, la generación redundante no puede limitarse sólo a un proceso de generación adicional y, cuando hay disponibles más de dos programas de generación, las ecuaciones booleanas pueden generarse de manera redundante en dos, tres o más conjuntos, por lo que dichas ecuaciones, por tanto el programa lógico de control y monitorización específico del sistema de estación puede comprobarse con un mayor nivel de seguridad, sin un aumento importante de costes o tiempos de procesamiento.

Según una mejora, puede proporcionarse una etapa preliminar en la que se genera la base de datos de entrada que contiene el diagrama del sistema de estación y la tabla de estado y se realiza una comprobación de la traducción de la tabla de corrección específica del programa y el diagrama de estación en el formato de entrada, para eliminar mediante filtrado las ecuaciones erróneas producidas por una tabla de estado e información del sistema de estación codificada de manera errónea, al lenguaje de la base de conocimiento para programas de generación. En este caso, la etapa preliminar para la generación específica de la estación del programa lógico de control y monitorización incluye las etapas de comprobar la base de conocimiento, tanto respecto a la estructura de la misma como respecto a la consistencia de los datos codificados en la base de conocimiento con el diagrama de sistema y con la tabla de estado.

En un ejemplo particular, las etapas anteriores se realizan de la siguiente manera:

La base de conocimiento se construye leyendo las definiciones y los datos contenidos en los diversos archivos de entrada de un “directorio de diagramas” y de un “directorio de estación”. Estas definiciones y datos corresponden al diagrama de estación expresado en un lenguaje codificado y a la base de datos de tabla de estado respectivamente. Una vez leídos, los datos y definiciones se añaden a la base de conocimiento, que se usa para realizar de manera apropiada las dos operaciones sucesivas.

Los dos programas de generación requieren los siguientes archivos de diagrama de entrada típicos

configurazione.pl

componenti.pl

subnet.pl

agenda.pl

ES 2 307 954 T3

Estos archivos deben distribuirse en un directorio, a continuación en el presente documento el “directorio de diagramas” al que pueden acceder los dos programas de generación. Además, este directorio debe contener un archivo (que acaba en una extensión “.pl”) para cada fase funcional referenciada en el archivo “agenda. pl”. Estas etapas funcionales son las definidas en el nivel de diagrama de sistema de estación.

Los dos programas de generación requieren los siguientes archivos de entrada, que se refieren a la base de datos obtenida a partir de la tabla de estado de cada estación:

db_tabella.pl

db2_tabella.pl

Estos archivos deben distribuirse en un directorio, a continuación en el presente documento el “directorio de estación” al que pueden acceder los programas de generación. Obviamente, este directorio puede ser diferente del “directorio de diagramas” definido anteriormente.

Tras el procesamiento, los programas de generación generan los siguientes archivos de informe, que se distribuyen en el “directorio de diagramas” y en el “directorio de estación” respectivamente de su programa de generación.

ades2++_schemistica.log

ades2++L_stazione.log

ades2_schemistica.log

ades2_stazione.log

En este caso, los dos programas de generación se denominan ades2 y ades2++ respectivamente.

Con respecto a los programas ades2++ o ades2 o ambos, los archivos anteriores contienen mensajes de texto que se refieren a las diversas etapas de ejecución de la aplicación, incluyendo cualquier mensaje de error generado por una sintaxis incorrecta de archivos de entrada o por errores durante la generación de ecuaciones booleanas específicas de la estación.

Por tanto, se generan las siguientes ecuaciones booleanas, para cada estación específica, en el siguiente archivo, que está contenido en el “directorio de estación”

ades2++_equazioni.dat

ades2equazioni.dat

El formato con el que el programa de generación ades2++ escribe ecuaciones booleanas también lo usa ades2. Se añadirán líneas de texto equivalente al inicio o al final del archivo, y se insertarán líneas de comentarios apropiadas para delimitar las ecuaciones producidas para cada etapa funcional. Si las ecuaciones se generan más de una vez, las dos últimas ecuaciones booleanas generadas se guardan en el “directorio de estación”, después de haberles dado un nuevo nombre adecuado como

ades2++_equazioni.bak

ades2_equazioni.bak

Partiendo de los datos de la base de conocimiento (siempre que estos últimos se generen correctamente), se genera una lógica de estación para cada etapa funcional tal como se define en el archivo “agenda.pl”. Esta lógica se genera como un conjunto ordenado de circuitos lógicos, construyéndose cada circuito lógico aplicando las definiciones relevantes del diagrama de principios a los datos específicos de la estación. Cada circuito contiene una red de componentes y una lista de uno o más componentes terminales.

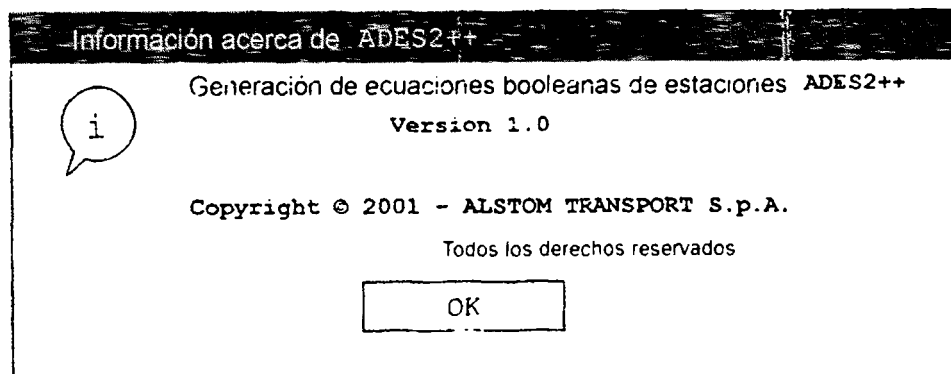
El programa para la generación redundante de ecuaciones booleanas ades2++ convierte los circuitos generados durante la etapa previa en ecuaciones booleanas. Cada circuito se convierte en una o más ecuaciones, determinándose también el número de ecuaciones generadas por ciertas restricciones de configuración impuestas por la unidad de control central, es decir, el denominado aparato informático esencial con base en la estación.

Cada ecuación está compuesta por una lista de variables booleanas resultantes y por una expresión compuesta por operaciones en términos que incluyen variables booleanas. Cada una de estas variables representa a su vez un

componente (terminal o no terminal) de un circuito, o un componente “virtual” que se usa para conectar dos ecuaciones construidas a partir del mismo circuito. El programa de generación escribe cada ecuación, en el orden apropiado, en el archivo denominado “ades2++_equazioni.dat”, que está contenido en el “directorio de estación” asociado a la estación seleccionada. En este archivo se generan ecuaciones exactamente en el mismo orden que el que tienen en el archivo equivalente, que se genera por el primer programa de generación ades2.

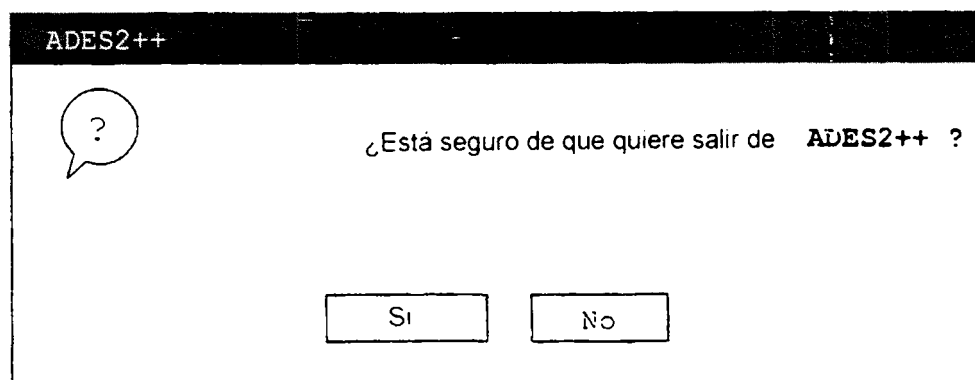
Ahora se describirá un ejemplo de interfaz de usuario, específicamente con referencia a la generación por el segundo programa de generación ades2++. Se proporcionará aquí una descripción de la etapa en la que se carga la base de conocimiento y se realiza la comprobación de consistencia y corrección, con referencia adicional a la etapa de generación que implica el primer programa de generación ades2 en el ejemplo anterior.

Cuando la aplicación ADES2++ se lanza en Windows, se muestra el siguiente mensaje de información general.



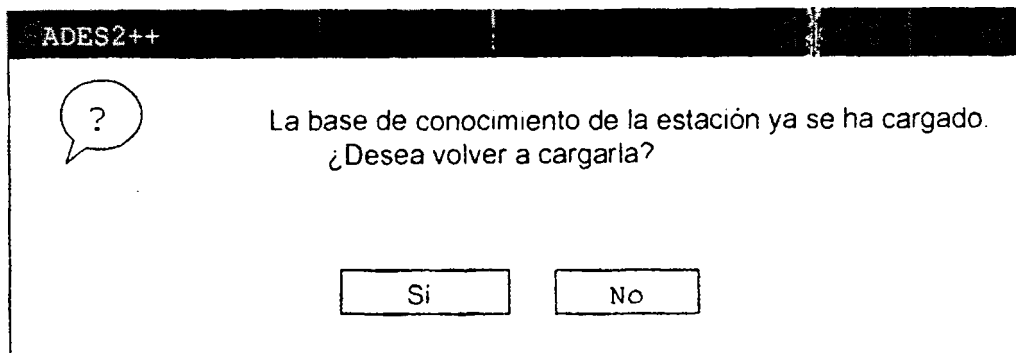
La pantalla del ordenador mostrará una ventana de aplicación, que contiene todos los controles y botones tal como se muestra en la ventana a continuación. Deberá indicarse que la versión de la aplicación se menciona en el título de la ventana. Como es habitual, la ventana puede moverse, minimizarse, maximizarse y cerrarse, utilizando las propiedades y botones típicos de Windows. Deberá indicarse además que la ventana muestra los archivos del directorio de diagramas y del directorio de estación que se usaron por el primer programa para generar el programa lógico de control y monitorización, denominado ades2.

La ventana contiene todos los controles que pueden usarse para seleccionar los directorios de diagramas y estación apropiados. En particular, la ventana contiene tres botones, usándose cada uno para seleccionar uno de los modos descritos anteriormente. También se proporcionan dos botones adicionales que permiten consultar archivos de informe de estación y diagrama respectivamente. La barra de estado en la parte inferior de la ventana se usa por la aplicación para mostrar cierta información de estado. Los botones siempre están habilitados, menos cuando una de las funciones principales está en ejecución. Esto permite al usuario usar la aplicación más de una vez, sobre el mismo conjunto de datos o sobre otros conjuntos de datos. El usuario puede abandonar la aplicación en cualquier momento, cerrando la ventana de aplicación. En este caso se pedirá al usuario que confirme que quiere salir, usando el diálogo siguiente.

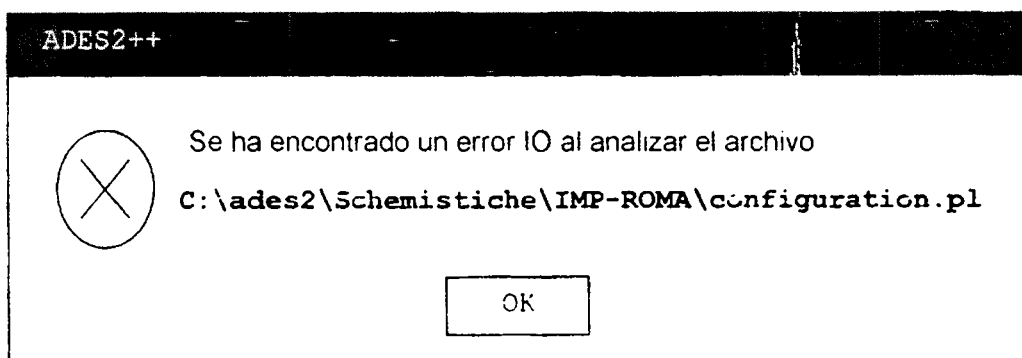


Para usar las propiedades anteriores, el usuario deberá rellenar los controles apropiados con el nombre completo de los directorios que contienen los archivos de entrada relativos a los diagramas y a la estación relevante.

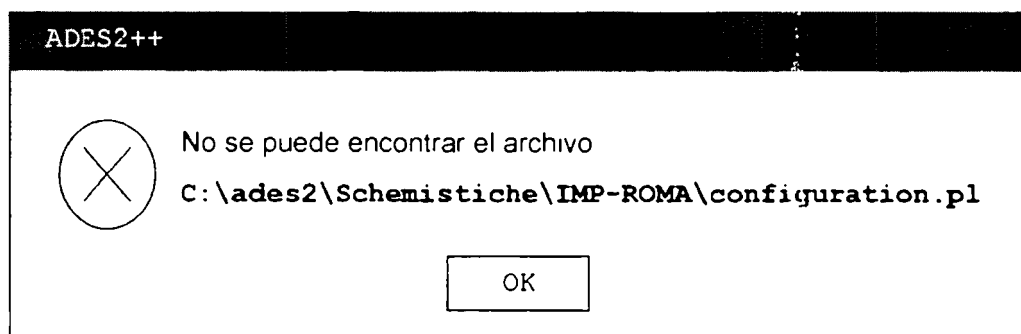
Haciendo clic con el botón izquierdo sobre “Carica Dati di Stazione” (cargar datos de estación), el usuario puede generar la base de conocimiento a partir de datos de diagrama y datos de estación específicos. Si ya se ha generado una base de conocimiento para los diagramas y la estación relevante, se mostrará el siguiente mensaje de aviso para pedir al usuario que confirme expresamente la nueva creación.



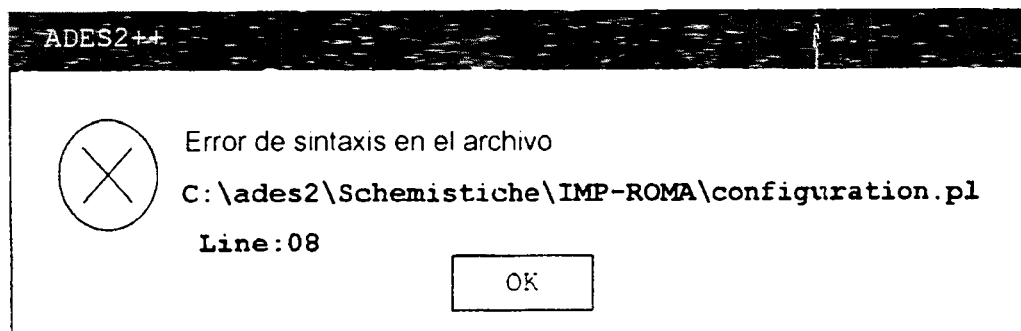
15 La propiedad de generación de la base de conocimiento intenta leer secuencialmente los archivos de entrada indicados. Si se produce un error de lectura de archivo, se muestra un mensaje como el mostrado a continuación y se termina la generación de la base de conocimiento.



35 Si uno de los archivos solicitados no se encuentra en los directorios especificados, se muestra un mensaje como el mostrado a continuación y se termina la generación de base de conocimiento.

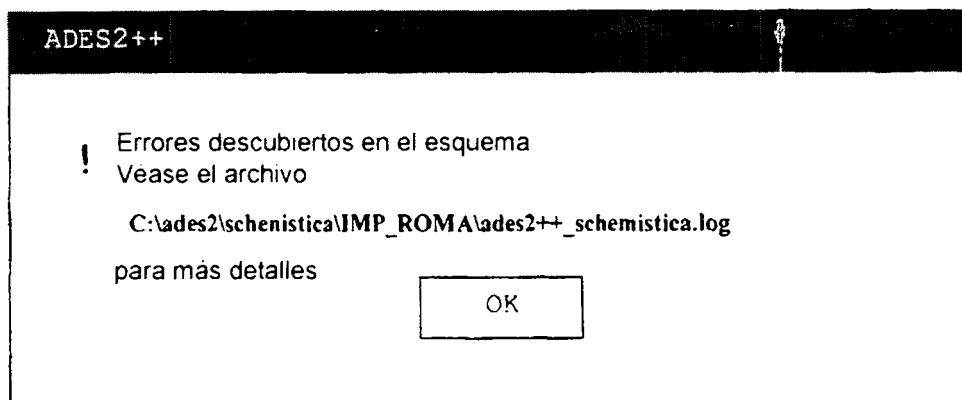


50 Además, si los archivos especificados contienen errores de sintaxis, la generación de la base de conocimiento terminará en cuanto se encuentra el primer error, tras lo que se muestra un mensaje indicando el nombre de archivo y el número de línea en el que se produjo el error (tal como se muestra a continuación).



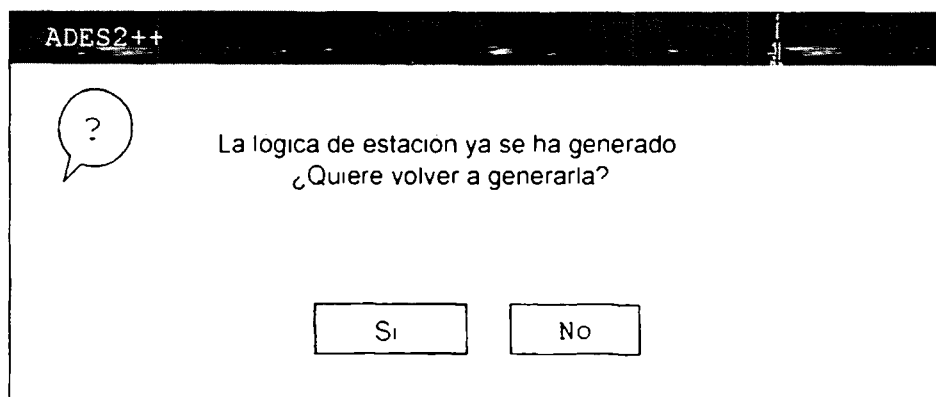
5 Sin embargo, si los archivos especificados no contienen errores de sintaxis, la generación de la base de conocimiento continuará hasta que se hayan leído todos los archivos de entrada. Cualquier otro error detectado en la definición del diagrama de principios, expresado en formato de entrada, se incluirá en el archivo de informe generado en el “directorio de diagramas”. No se insertará ninguna definición inconsistente en el diagrama de principios en la base de conocimiento. Sin embargo, de cualquier modo se cargarán definiciones incompletas del diagrama de principios, tales como reglas de diseño asociadas a componentes inexistentes.

10 Si se producen errores, se mostrará un mensaje relevante al final del proceso de generación, tal como se muestra a continuación.



30 Haciendo clic con el botón izquierdo sobre “Generate Station Logic” (generar lógica de estación), ADES2++ ejecutará la función de generación de la lógica de estación, a partir de la base de conocimiento generada previamente. (Si la base de datos está incompleta, la lógica generada también estará incompleta).

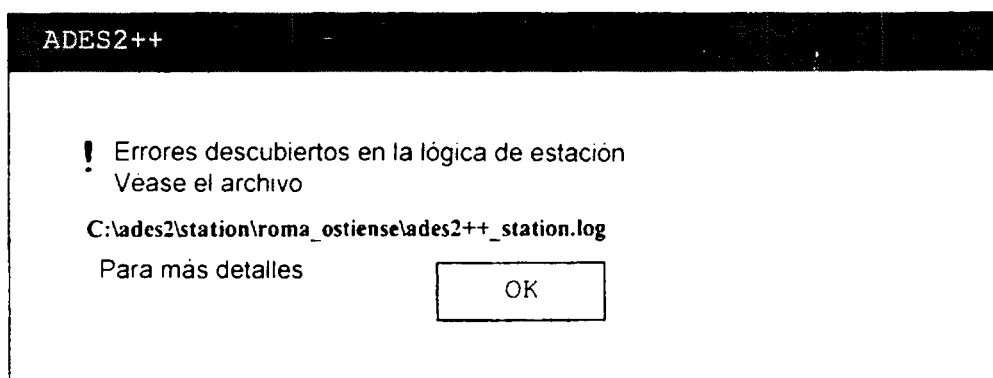
35 Si ya se ha generado la lógica de diagrama y de estación, antes de volver a generarlos, se pedirá una confirmación al usuario, mediante el mensaje siguiente.



55 Mientras que se ejecuta la generación de la lógica, aparecerán mensajes apropiados en la barra de estado, para indicar la etapa funcional en la que el sistema está generando la lógica, y el número de circuitos generados (para esa etapa), así como el número total de circuitos generados hasta ese momento.

60 El número total de circuitos generados (para todas las etapas) se mostrará en la barra de estado cuando se haya completado el proceso. Deberá indicarse que el número de circuitos generados puede ser menor que el número de ecuaciones que se generarán a continuación durante la etapa de almacenamiento.

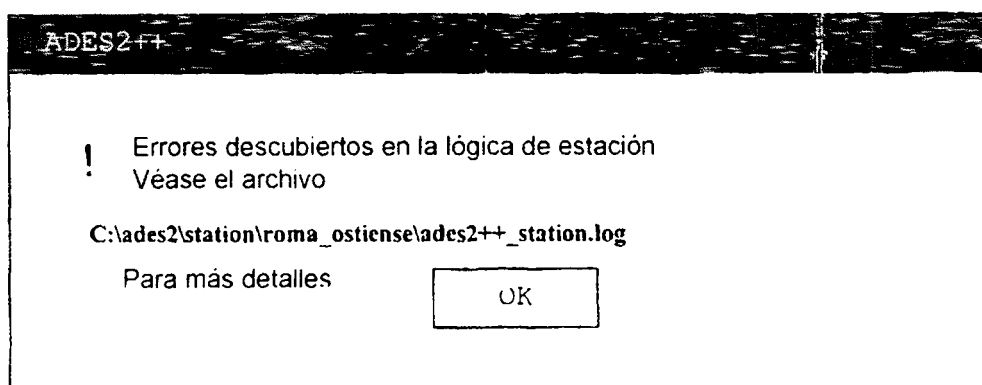
65 Durante la etapa de generación lógica, dependiendo de los datos cargados actualmente, un único componente puede asignarse como un componente terminal a más de un circuito lógico. Cada vez que se produzca tal evento se identificará como error, y como tal se indicará en el archivo de informe generado en el “directorio de estación”. Incluso cuando el nombre de un componente supere la máxima longitud admitida, se mostrará un mensaje. Si se han detectado uno o más errores al final de la etapa de generación de la lógica, aparecerá un mensaje de aviso apropiado en la pantalla, tal como se muestra a continuación.



Haciendo clic con el botón izquierdo sobre “store station equations” (guardar ecuaciones de estación), los circuitos lógicos generados se convierten en ecuaciones booleanas. (Cuando no se produzca generación lógica se generará un archivo vacío, es decir, sin ecuaciones).

Cuando previamente se haya generado un archivo de ecuaciones para la misma estación, se creará una copia de seguridad del mismo antes de iniciarse la nueva generación de ecuaciones.

Puede ocurrir que, mientras se generan resultados a partir de una cierta ecuación, la aplicación intente usar un componente que previamente se haya definido como “estado”, pero que no esté usándose como un componente no terminal en ningún circuito. Este evento se notificará por la aplicación. Si ningún otro componente terminal se asigna a este circuito, no se generará ninguna ecuación booleana del mismo. En este caso, la aplicación mostrará un mensaje de aviso al final del proceso de generación, tal como se muestra a continuación.



Como es habitual, estos eventos también se presentarán en el archivo de informe creado en el “directorio de estación”.

Haciendo clic con el botón izquierdo sobre uno de los dos botones “Open Scheme report” (abrir informe de esquema) o “Open Station Report” (abrir informe de estación), el usuario puede llamar de nuevo los archivos de informe generados relativos al diagrama de principios expresado en formato de entrada o a los datos de estación. En otras palabras, haciendo clic sobre uno de estos dos botones, el usuario puede abrir una ventana de texto que informe acerca del contenido actual de los dos archivos.

El usuario puede abrir más ventanas de informe para el mismo archivo.

Con referencia al ejemplo anterior, pueden resultar diferencias entre los dos programas de generación en los procedimientos de comprobación de la consistencia de los datos de entrada y en los mensajes de error durante la generación de la lógica.

Un archivo de ecuaciones generado por ades2++ para un cierto diagrama y una cierta estación puede compararse directamente con el archivo de ecuaciones generado por ades2 a partir de los mismos archivos. Por tanto, pueden usarse herramientas de comparación comerciales para comparar los dos archivos.

Particularmente, el número y orden de las ecuaciones, y los resultados de cada ecuación en un archivo deben ser idénticos a los del otro archivo. Teas expresiones de ecuación de un archivo también deberán ser equivalentes a las del otro archivo, es decir, cada término y operación contenidos en la expresión de una ecuación en un archivo también deberán aparecer en la expresión de la ecuación asociada en el otro archivo. El orden de términos en cualquier expresión de suma o producto de una expresión en un archivo puede ser diferente del de la expresión asociada en el otro archivo. Esto se debe al hecho de que los algoritmos que se usan para construir las expresiones son diferentes de manera intencionada en las dos aplicaciones y, aunque ambos tienen que cumplir los estrictos requisitos de generación de lógica de estación (es decir, tienen que ser completos y estar expresados en el orden correcto), habrá casos en los que diferentes requisitos implicarán diferencias en el orden de las expresiones. En cualquier caso, estos casos serán muy poco frecuentes en la práctica.

Con respecto a los programas de comparación, este método tiene la ventaja de que pueden usarse programas comerciales, tales como: MKS Visual Difference para Win32 - Versión 3.2b Mortice Kern Systems Inc y/o Microsoft® WinDiff - Versión 4.0 Microsoft Corp.

La figura 2 muestra un aparato estacionario informático esencial, es decir, una unidad de monitorización y control central, según la presente invención, que también integra los medios para la generación redundante de programas lógicos de control y monitorización específicos del sistema de estación.

El número de referencia 1 indica una estación que tiene una pluralidad de diferentes elementos 101 de estación de 1 a N, tales como luces de señalización, agujas de vía férrea, circuitos de vía, y otros.

Cada elemento 101 está controlado por un controlador que puede estar consistir en o incluir hardware o software, y es específico del elemento y siempre el mismo pala cada elemento específico. Los controladores 2 tienen interfaces de entrada para controles e interfaces de salida para señales de diagnóstico y retroalimentación. Estas entradas y salidas están conectadas con entradas y salidas apropiadas de una unidad 3 de control central que se denomina aparato estacionario informático esencial.

Esta unidad 3 central incluye programas de gestión, para controlar y monitorizar los elementos 101 así como programas de diagnóstico, y también constituye la interfaz entre el personal y el sistema.

Desde un punto de vista funcional, la unidad central puede estar dividida en dos áreas principales. Una de ellas, indicada con el número de referencia 103 en la figura 2, está diseñada para ejecutar procedimientos de diagnóstico, monitorización de elementos y control de elementos, y está compuesta por programas orientados al procedimiento aplicables de manera universal.

La otra área, indicada con el número de referencia 203 en la figura 2, constituye la lógica de control y monitorización real y consiste en un programa lógico de control y monitorización. Este programa también puede gestionar posiblemente funciones de diagnóstico, aunque generalmente se proporciona una sección especial para el diagnóstico.

Las dos áreas 103, 203, que están separadas de manera sistemática, deben coexistir, de lo contrario el sistema no podrá funcionar. Los programas de gestión generales de diagnóstico, control y monitorización deberán estar integrados o establecer de cualquier modo una interfaz con la lógica de control y monitorización. Esta última depende estrictamente de e incorpora todas las peculiaridades y especificidades del sistema de estación y de las reglas de gestión del tráfico de la vía férrea que pueden aplicarse al mismo. Por tanto, la lógica de estación deberá generarse de manera que esté dedicada y sea específica de cada estación a la que está asociada la unidad 3 central. Según la invención, el aparato estacionario informático esencial, es decir la unidad 3 de control y monitorización incluye medios para la generación automatizada del programa lógico de control y monitorización que están integrados de manera estable, tal como la sección 303, en el sistema o software de dicha unidad 3 de control.

Particularmente, estos medios consisten en medios para introducir el diagrama 4 de estación y las reglas 5 de cambio de estado para los diversos elementos para la gestión del tráfico de la vía férrea en dicha estación, y medios para generar una base de conocimiento a partir de dicha información, que deberá usarse por un programa para generar dicho programa lógico de control y monitorización. En la realización ilustrada, dichos medios consisten en medios de hardware, es decir un ordenador dedicado o por el ordenador que también controla la unidad 3 central de monitorización y control, y en el software cargado en el mismo.

Particularmente, dicho software está diseñado de modo que garantiza una generación 7, 8 redundante de programas lógicos de control y monitorización y ejecuta posteriormente una comprobación en la sección 303 de generación y/o la unidad 3 central de monitorización y control, sobre los programas lógicos generados, basándose en una comparación 6 de identidad entre los múltiples, particularmente dos, programas 7, 8 lógicos generados en paralelo. La generación paralela se realiza según dos programas de generación diferentes que recuperan datos de la misma base 4, 5 de conocimiento y proporcionan las ecuaciones booleanas diseñadas para formar el núcleo de los algoritmos de los programas lógicos de control y monitorización. Si la comparación da como resultado la identidad entre los dos conjuntos de ecuaciones booleanas proporcionados por los dos programas 7, 8 de generación diferentes, o que tienen un cierto grado de diferencia, entonces se considera que dicho conjunto de ecuaciones booleanas es correcto y se usa para generar el programa lógico de control y monitorización en su forma completa, lo que obviamente requiere secciones de adaptación a las restricciones estructurales impuestas por la construcción de la unidad 3 de monitorización y control.

ES 2 307 954 T3

Deberá indicarse que la generación redundante de la lógica 203 de control y monitorización no está limitada a dos procedimientos de generación paralelos, y que también pueden proporcionarse tres o más procedimientos de generación paralelos.

- 5 Añadiendo de forma permanente la sección 303 para generar la lógica de control y monitorización a la unidad 3 central de monitorización y control, la unidad 3 central de monitorización y control puede modificarse e integrarse fácilmente, siempre que se hagan cambios en el sistema de estación de vía férrea, por ejemplo se añadan o eliminen elementos. En este caso, la sección 303 para generar la lógica de control y monitorización sólo se usaría para hacer un cambio de la lógica de control y monitorización usada previamente para tener en cuenta los cambios de sistema.
- 10 No sólo pueden requerirse cambios por la adición o eliminación de elementos que deben controlarse, sino también por cambios en reglas de control y monitorización de elementos, que se resumen en las denominadas tablas de estado. En este caso, la lógica de control y monitorización también necesita cambiarse.

- 15 La ventajas de la comprobación de la corrección y la generación redundante por la comparación entre los programas, es decir, los conjuntos generados de ecuaciones booleanas, resultan particularmente evidentes cuando se hacen cambios en el sistema. En este caso, mientras que en la técnica anterior la lógica modificada debía comprobarse normalmente *in situ*, gracias al método de la invención, todo se procesa por el ordenador de la unidad central o por una estación secundaria con base en el ordenador. Esto reduce drásticamente los tiempos de actualización del sistema, así como los costes.

- 20 Aunque la invención se ha descrito con referencia particular a lógicas de control basadas en software, debe indicarse que también puede aplicarse cuando debe implementarse una lógica de control en hardware dedicado. En este caso, por ejemplo redes de componentes lógicos en forma de relés o componentes semiconductores sustituirían el software lógico de control y monitorización, generándose diagramas de circuito directa y automáticamente por programas de generación.

- 25 Además, la descripción anterior muestra claramente que la etapa de generar una lógica de control y monitorización de software deriva directamente de la etapa de generar circuitos lógicos virtuales, traducidos además por el programa de generación a la forma de software, cuyo núcleo está formado por conjuntos de ecuaciones booleanas.

30

35

40

45

50

55

60

65

REIVINDICACIONES

1. Método para generar unidades de control lógicas para aparatos informáticos esenciales con base en estaciones ferroviarias, es decir, para unidades de control de sistema de estaciones ferroviarias, que comprende al menos un ordenador esencial que, basándose en un programa de control que funciona en combinación con una unidad lógica, envía controles de cambio de estado a denominados elementos de maniobra, es decir, dispositivos que están diseñados para realizar operaciones específicas relacionadas con la circulación de trenes, tales como dispositivos de señalización y/o agujas de vía férrea y/o circuitos de vía, o similares, y recibe señales de diagnóstico y/o retroalimentación de estado desde dichos elementos de maniobra, generándose dicha unidad lógica automáticamente por un programa de generación, basándose en las condiciones del entorno tal como se define por un diagrama de estación, que comprende una lista de elementos de maniobra, y por una tabla de estado, en la que se fijan reglas de adopción de estado y/o cambio de estado para dichos elementos de maniobra, con referencia al estado y/o cambio de estado de otros elementos de maniobra y/o a la gestión apropiada del tráfico de la vía férrea, siendo dicha unidad lógica una red de circuitos con componentes que funcionan según funciones lógicas booleanas y estructurados de manera apropiada conforme al diagrama de estación y a la tabla de estado, o siendo dicha unidad de control lógica un programa que incluye algoritmos compuestos por funciones lógicas booleanas, que funcionan como redes de circuitos lógicos booleanos, **caracterizado** porque incluye una etapa para comprobar la corrección de la unidad lógica generada automáticamente, etapa de comprobación que incluye las etapas siguientes:
 - generar paralelamente dos unidades de control lógicas, según el mismo diagrama de estación y la misma tabla de estado, generándose cada unidad por uno de dos programas de generación que son lo más diferentes posible entre sí;
 - comparar las redes de circuitos lógicos entre sí o los programas lógicos de simulación de redes proporcionados por los dos programas de generación diferentes para comprobar diferencias estructurales entre los mismos.
2. Método según la reivindicación 1, **caracterizado** porque cuando se consigue un resultado de identidad, se considera que se ha comprobado la corrección de las redes de circuitos lógicos o del programa lógico generado.
3. Método según la reivindicación 1 ó 2, **caracterizado** porque cuando se descubre que los dos programas lógicos no son idénticos, se realiza una etapa de comprobación de errores, y se repiten las etapas de generación paralela de las redes de circuitos lógicos y/o programas lógicos virtuales de simulación de redes.
4. Método según una o más de las reivindicaciones anteriores, **caracterizado** porque la diferencia entre los dos programas de generación se refiere a sus lenguajes o a los entornos de programación en los que están escritos.
5. Método según una o más de las reivindicaciones anteriores, **caracterizado** porque los dos programas de generación diferentes usan diferentes algoritmos de generación.
6. Método según una o más de las reivindicaciones anteriores, **caracterizado** porque los dos programas de generación diferentes son dos redes neuronales diferentes.
7. Método según una o más de las reivindicaciones anteriores, **caracterizado** porque incluye una etapa para preparar una base de conocimiento que contiene datos relacionados con un diagrama de estación y datos relacionados con una tabla de estado que están codificados de manera que son comprensibles por ambos programas de generación.
8. Método según la reivindicación 7, **caracterizado** porque uno o ambos programas de generación incluyen una etapa de generación previa, en la que se comprueba la consistencia y corrección de los datos de la base de conocimiento tanto respecto al significado como a la estructura de los datos.
9. Método según una o más de las reivindicaciones anteriores, **caracterizado** porque incluye un programa para comparar los programas lógicos y/o las redes de circuitos lógicos generados por los dos programas de generación, programa de comparación que es independiente de los programas de generación.
10. Método según una o más de las reivindicaciones anteriores, **caracterizado** porque los dos programas de generación generan los programas lógicos con el siguiente procedimiento:
 - generar redes de circuitos lógicos que usan componentes de hardware lógicos;
 - convertir las redes de circuitos lógicos generadas de este modo en algoritmos lógicos compuestos por conjuntos de ecuaciones booleanas cuyo comportamiento corresponde al de dichas redes de circuitos lógicos.
11. Método según una o más de las reivindicaciones anteriores, **caracterizado** porque se usa cuando deben cambiarse circuitos lógicos y/o programas lógicos para su adaptación a cambios del diagrama del sistema de estación y/o de la tabla de estado.
12. Aparato estacionario informático esencial que incluye un ordenador en el que se carga un programa para controlar y monitorizar elementos de maniobra de un sistema de estación, que funcionan según diferentes reglas,

ES 2 307 954 T3

caracterizado porque dicho programa incluye una sección de programas generales orientados al procedimiento, que pueden aplicarse tanto a la estructura del sistema de la estación como a la tabla de estado, estableciendo dicho programa una interfaz y estando integrado con un programa lógico de control y monitorización, que incorpora la estructura del sistema de la estación y la tabla de estado, y se genera y comprueba automáticamente por una sección para generar el programa lógico de control y monitorización del aparato estacionario informático esencial, estando adaptada esa sección para generar el programa lógico de control y monitorización para llevar a cabo el método según uno o más de las reivindicaciones 1 a 11.

13. Aparato estacionario informático esencial según la reivindicación 12, **caracterizado** porque la sección para generar el programa lógico de control y monitorización constituye una sección para cambiar y/o actualizar dicho programa lógico de control y monitorización.

14. Aparato informático esencial con base en la estación según la reivindicación 12 ó 13, **caracterizado** porque la sección para generar el programa lógico de control y monitorización comprende al menos dos programas de generación diferentes, para generar programas lógicos de control y monitorización comparables que se cargan, después de una comprobación de identidad con éxito, en la memoria del aparato estacionario informático esencial y establecen una interfaz con la sección de programas generales orientados al procedimiento.

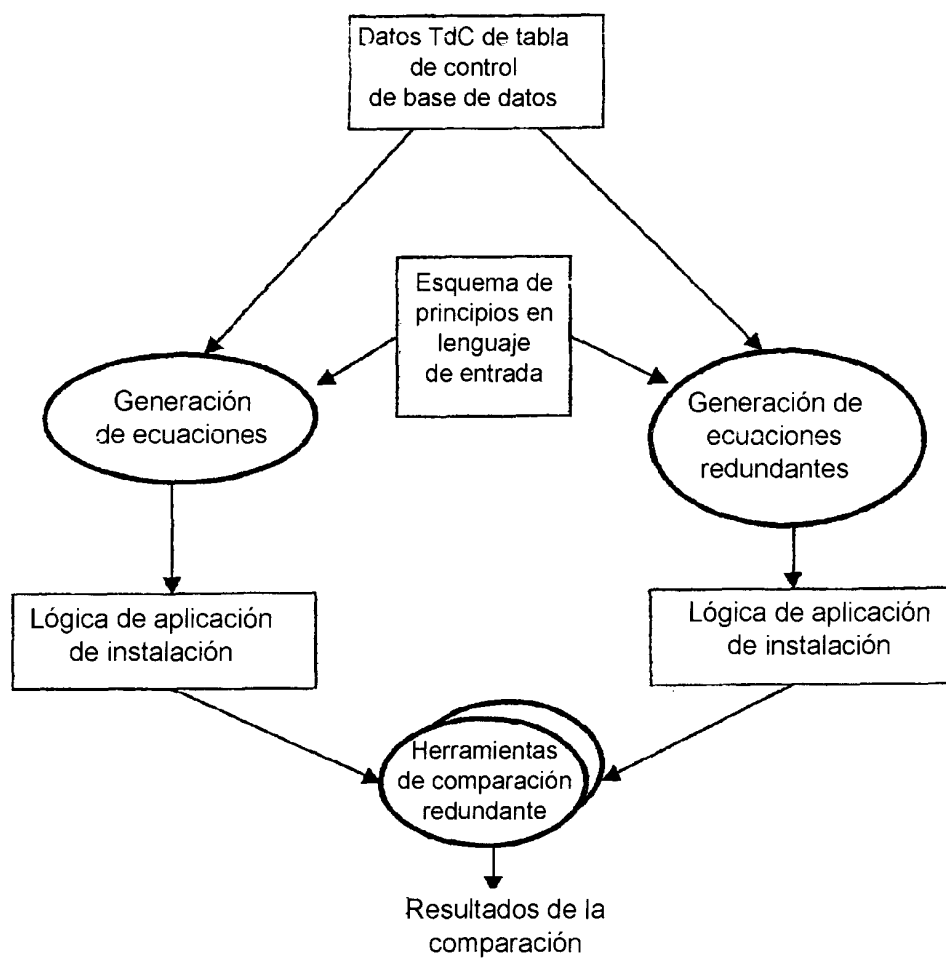


FIG. 1

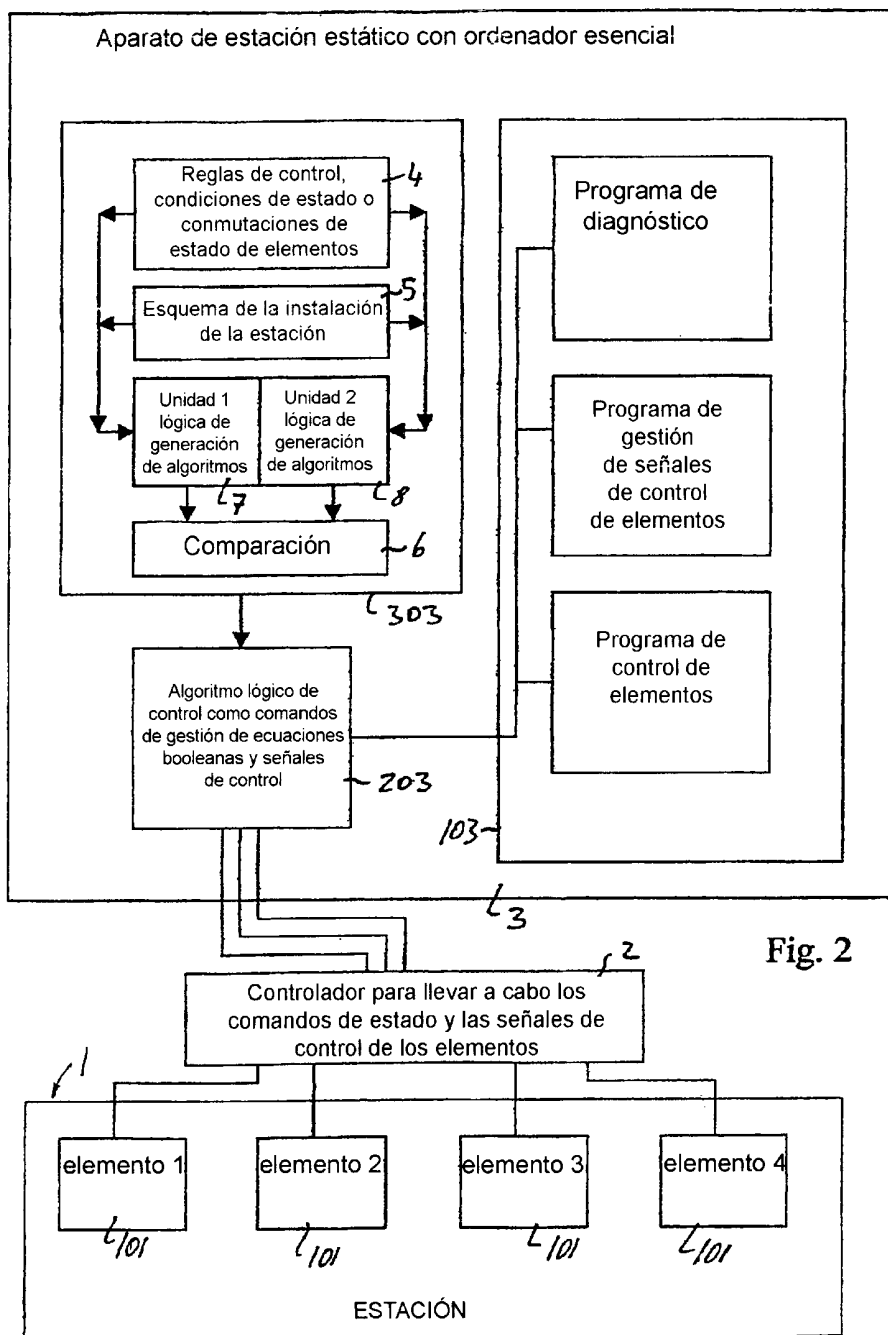


Fig. 2