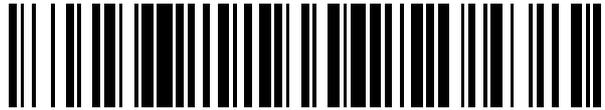


19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 457 494**

51 Int. Cl.:

H04L 12/24 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **17.02.2011 E 11706202 (6)**

97 Fecha y número de publicación de la concesión europea: **01.01.2014 EP 2543160**

54 Título: **Procedimiento y sistema de gestión de operaciones en un terminal de telecomunicaciones con máquina de estados**

30 Prioridad:

05.03.2010 US 310769 P

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

28.04.2014

73 Titular/es:

**TELFÓNICA, S.A. (100.0%)
Gran Vía, 28
28013 Madrid, ES**

72 Inventor/es:

**VILLOSLADA DE LA TORRE, EDUARDO;
MARTÍNEZ ELICEGUI, JAVIER y
ORTEGA BARRADO, PEDRO JOSÉ**

74 Agente/Representante:

CARPINTERO LÓPEZ, Mario

ES 2 457 494 T3

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

Procedimiento y sistema de gestión de operaciones en un terminal de telecomunicaciones con máquina de estados

Campo de la invención

5 La presente invención se refiere al campo de las telecomunicaciones y, más en particular, se refiere al campo de los terminales de servicio conectados a un servidor remoto.

Estado de la técnica

Concretamente, un Punto de Venta (TPV) (en inglés Point of sale POS) es un sistema que gestiona el proceso de venta mediante una interfaz accesible para vendedores, pero en términos generales se aplica a cualquier aplicación desde la que se facilita un proceso de venta. Por ello se puede realizar una clasificación:

- 10
- Aplicaciones completas para PC, cuyas funciones básicas son típicamente la creación e impresión del recibo, ticket o factura de venta, que comprende las referencias y precios detallados de los artículos vendidos, actualizar el nivel de existencias y mercancías en la base de datos y permitir la autorización para el pago con tarjetas de crédito/débito que posteriormente es transferida a las entidades bancarias.
- 15
- Puede haber enfoques compactos, donde todos los elementos necesarios están integrados en el terminal (CPU, impresora, pantalla, teclado) en una sola máquina, así como enfoques modulares, basados en un PC convencional, donde se conectan los distintos periféricos, junto con un software instalado sobre un sistema operativo convencional, lo cual permite el uso de funciones típicas de un PC.
- 20
- Datafonos, dispositivos de pequeño tamaño, basados en un teclado reducido, varios tipos de lector de tarjetas (banda magnética, tarjeta inteligente, sin contacto) y un software de aplicación que se comunica con la parte servidora remota.
- 25
- *WebPOS*, donde el acceso por un navegador de internet y una aplicación *Web* a los procedimientos necesarios para realizar la venta está garantizado, típicamente con cargo a una tarjeta de crédito/débito.
 - *mPOS (mobile POS)*, en donde el teléfono se convierte en el dispositivo para realizar las transacciones de venta. Pueden ser variantes de los casos anteriores: un *webPOS* que se ejecuta en el teléfono o añadiendo un componente capaz de leer cualquier tipo de tarjeta, actuando como un datafono.

Dentro de esta clasificación, centrándose en los datafonos, el modelo de operación actual es que las aplicaciones se ejecutan en el dispositivo y cuando finaliza la operación se envían los datos de la transacción al servidor con el que se comunica, para consolidar los datos, utilizando para ello cualquier tecnología de transmisión de datos.

30 El panorama de este tipo de dispositivos se caracteriza por la diversidad de fabricantes y porque los desarrollos son realizados a medida por cada fabricante: no son sistemas abiertos en el que quien quiera pueda hacer desarrollos, no es habitual que para un TPV puedan realizarse fácilmente desarrollos por terceros.

35 Han surgido varias iniciativas para lograr la estandarización de los dispositivos, de forma que, con independencia del fabricante, los dispositivos expongan un API común y los desarrollos puedan trasladarse de forma transparente de un dispositivo a otro, de un fabricante a otro, sin ningún problema. Dentro de estas iniciativas destacan *STIP (Small Terminal Interoperable Platform)*, de *GlobalPlatform* [<http://www.globalplatform.org/>] o *UnifiedPOS* [<http://www.nrf-arts.org/UnifiedPOS/>]

40 *STIP* es el acrónimo de *Small Terminal Interoperability Platform* ("Plataforma de interoperabilidad de terminales pequeños"). El consorcio *STIP* es un grupo de proveedores de soluciones de transacciones seguras, lo cual incluye, fabricantes de terminales, fabricantes de tarjetas inteligentes y otros. Fue formado para definir una especificación Java para terminales pequeños y dispositivos orientados a transacciones.

45 En la figura 1 se muestra la arquitectura subyacente a la tecnología *STIP*. La plataforma 11 expone a través de sus interfaces software 15 las capacidades de interacción sobre ella de los diferentes elementos que la componen 13, tales como la impresora, el lector de banda magnética, lector de tarjetas sin contacto ... Una aplicación 12 incluye un elemento principal 16 que controla la operación de la misma, así como un conjunto de servicios intermedios 14, que permitan interactuar con la plataforma 11 y sus elementos 13.

Las interfaces 15 expuestas tanto por los elementos 13 de la plataforma como por los servicios intermedios de la aplicación 14 siguen un modelo en el que los servicios generan eventos ante cambios en el estado de los componentes, mientras que la aplicación realiza peticiones a los servicios 14 para controlar su operación.

El objetivo de *STIP* es especificar una plataforma software que proporcione lo siguiente:

- 50
- soporte para múltiples aplicaciones de transacciones seguras en un terminal,
 - interoperabilidad para que las aplicaciones puedan ser ejecutadas en un amplio rango de dispositivos,
 - un procedimiento de gestión del ciclo de vida de aplicación que pueda ser implementado en dispositivos de pequeño tamaño, con recursos limitados, algo habitual en el entorno de dispositivos de tarjetas.

La tecnología STIP satisface los requisitos funcionales anteriores a través de las siguientes características:

- Lenguaje de programación de alto nivel común: la base de interoperabilidad de aplicaciones en diferentes plataformas hardware es utilizar un lenguaje de programación común con independencia del hardware subyacente. La solución de STIP confía en el uso de lenguajes orientados a objetos como Java. Además, la tecnología STIP ha definido un subconjunto de la base más común del API de Java para proporcionar un subconjunto común utilizable en todas las plataformas, independientemente de la versión del lenguaje implementado.
- Definición de los accesos a los recursos y periféricos comunes en terminales pequeños de forma portable: el acceso a todos los recursos se realiza mediante controles de servicio.

La principal dificultad es proporcionar un API flexible que permita varias configuraciones de periféricos y, al mismo tiempo, no comprometer e incluso reforzar la seguridad y la interoperabilidad. La solución se apoya en el siguiente enfoque: cada posible recurso de la plataforma (periféricos, almacenamiento...) es considerado como un servicio 14, que es implementado por una biblioteca software si está presente en la plataforma. El API de STIP no proporciona ningún acceso directo a esas bibliotecas.

Por otra parte, una aplicación puede, únicamente, acceder a un interfaz de control de servicio 15 estandarizado para cada servicio. Para acceder y utilizar un servicio, la aplicación debe solicitar la apertura de un canal de comunicación entre el control del servicio que maneja y el servicio real escondido tras él.

Además, para obtener un objeto de control de servicio de un tipo concreto, la aplicación debe primero solicitar al gestor de control de servicios un objeto de control de servicio del mismo tipo. Hay ciertas ventajas que surgen al utilizar este enfoque:

- Cuando un tipo particular de servicio no está presente en una plataforma determinada, no es necesario que la plataforma implemente el control de servicio relacionado. La plataforma STIP sólo define la declaración del interfaz del control de servicio pero no su implementación. De este modo, la interoperabilidad es posible sin sacrificar la flexibilidad.
- Las aplicaciones no tratan con APIs de servicio específicas sino con API de control de servicio estandarizados. Esto es importante para la interoperabilidad puesto que las bibliotecas pueden ser específicas de la plataforma pero el interfaz puede ser común para todas las plataformas.
- La seguridad se maneja de una forma cómoda por parte de la plataforma, puesto que no se puede acceder a ningún recurso sin dos solicitudes específicas desde la aplicación. Estas solicitudes se utilizan para obtener la instancia del control de servicio y para abrir a través de este control de servicio un canal de comunicación con el servicio específico. De este modo, las implementaciones de servicios reales están automáticamente protegidas por la plataforma.

Resumiendo, STIP satisface las necesidades de flexibilidad, seguridad e interoperabilidad.

El enfoque de STIP se sustenta en el uso sistemático de un estilo de programación basado en eventos. La maquinaria subyacente para solicitar eventos es simple, completamente especificada e independiente de la implementación. Esto mejora la programación de aplicaciones altamente reactivas a la vez que refuerza la seguridad e interoperabilidad de aplicaciones.

El modelo de operación habitual es que las diferentes opciones de ejecución se encuentran en la aplicación existente en el propio dispositivo, y para actualizar la aplicación es necesario realizar un mantenimiento dispositivo por dispositivo. Este mantenimiento, bien se realice de forma remota o de forma local, es necesario hacerlo para cada dispositivo, con una complejidad en el proceso de actualización y un incremento en coste de acuerdo con la planta instalada.

Por ello es de especial interés simplificar la actualización de las aplicaciones que se ejecutan en el TPV. En esta línea el camino tomado ha sido trasladar la lógica de las aplicaciones desde el TPV al servidor, de tal forma que en vez de conectarse al servidor únicamente al finalizar la transacción, el TPV se conecta al servidor en pasos intermedios, con lo que el TPV puede ser más simple y se pueden implementar aplicaciones con una flexibilidad que de otra forma no sería posible. Esta es la idea fundamental de la patente US 5696909.

En la actualidad se han propuesto aproximaciones relacionadas que trasladan el modelo de aplicaciones *web* para Internet a la operación para un TPV; de esta forma, el TPV se convierte en un *browser*, con capacidades para interpretar un lenguaje de marcas y gestionar los recursos del dispositivo, realizando peticiones para descargarse las nuevas páginas según se necesiten.

Estas propuestas suponen un avance en la idea de trasladar el control de la aplicación al servidor pero siguen presentando limitaciones:

La patente US 5696909 establece un modelo general basado en crear un elemento intermedio que asuma parte de las capacidades hardware y software del terminal TPV y un conjunto de transacciones predeterminadas. Esto no permite realmente un control completo desde el servidor de las aplicaciones que se ejecutan en el terminal TPV. Además, la interacción frecuente con el servidor puede no ser asumible en escenarios en los que las comunicaciones planteen

problemas, bien por las prestaciones (tiempos de respuesta), bien por el coste de las mismas.

En este aspecto, una de las mencionadas propuestas ya existentes implica una fuerte restricción de que los TPV sean capaces de interpretar su lenguaje de marcas, lo que implica una importante capacidad de cálculo que supone una limitación para terminales TPV básicos.

- 5 En resumen, el problema técnico objetivo que hay que resolver es la optimización de la gestión de terminales TPV o terminales de servicio mediante el uso de un servidor que permite que los terminales de servicio descarguen programas de las aplicaciones y considerar las peculiaridades de estos terminales cuando se utilizan los programas descargados.

10 El documento US 2010 / 058329 divulga un procedimiento para la gestión de dispositivos de terminal en una red descargando un sistema operativo (SO) y configurando el SO descargado verificando toda una cadena de condiciones, estados del SO, y transiciones entre dichos estados. Sin embargo, la descarga de los programas de SO no se desencadena por una solicitud de un terminal de servicio, sino que la operación de descarga es iniciada por un servidor, que posee una nueva versión del SO. La configuración del sistema operativo descargado se lleva a cabo en el terminal usando una máquina de estado para verificar toda la cadena de condiciones, estados y transiciones mencionada anteriormente, pero esta configuración no considera ninguna interacción entre los terminales y los usuarios y no hay 15 entradas de usuario para configurar los módulos de los terminales (de hecho, el documento US 2010 / 058329 no considera la posibilidad de gestionar un terminal de servicio que tiene una pluralidad de módulos que a configurar).

20 El documento US 2004/107277 divulga un procedimiento para descargar programas de ordenador desde un servidor y para configurar automáticamente terminales que soliciten la descarga de un programa específico. El servidor envía los programas solicitados a sus terminales, pero sin tener en cuenta ninguna cadena de condiciones, estados de programa o transiciones entre los estados de programa. La configuración de los terminales no considera ninguna interacción entre terminales y usuarios, ni tampoco tiene en cuenta entradas del usuario, ni atenciones de terminales con múltiples módulos a configurar.

Sumario de la invención

25 La idea fundamental de la presente invención es ubicar una máquina de estados dentro del propio terminal de servicio o TPV de tal forma que sea posible descargar desde el servidor los algoritmos que deben ejecutarse para cada aplicación. Puesto que los algoritmos residen en el servidor y son descargados al terminal de servicio, el servidor puede controlar en todo momento la operativa que se realiza en cada terminal de servicio.

30 La máquina de estados se apoya en STIP, en donde cada elemento del terminal de servicio lleva asociado un módulo que lo controla y es capaz de generar los eventos correspondientes a cada cambio de estado. De esta forma, se convierte en una máquina de estados genérica, en dónde se pueden incorporar nuevos periféricos, sin cambios en la máquina de estados.

El modelo de operación es que cada evento generado por un elemento del terminal de servicio o TPV, junto con toda la información asociada al evento, son recogidos por la máquina de estados para que determine la siguiente acción a realizar, en base al algoritmo que se haya descargado previamente del servidor.

35 Con este modelo de operación, se minimizan el número de peticiones al servidor sin perder el control por parte del servidor, en base al algoritmo descargado en el terminal de servicio o TPV. Por tanto, al tener menos transacciones con el servidor, con este modelo, no son necesarios tiempos de respuesta en las comunicaciones muy pequeños y el volumen de datos intercambiados también se reduce, manteniendo en todo momento la flexibilidad de las aplicaciones a ejecutar. De esta forma se superan totalmente las limitaciones de las soluciones actuales.

40 En un primer aspecto de la presente invención, se proporciona un procedimiento para gestionar una operación en un terminal de servicio a través de un servidor remoto, en el que dicho terminal de servicio comprende una pluralidad de módulos, una máquina de estados, configurada para cargar algoritmos correspondientes a aplicaciones y ejecutar dichas aplicaciones, y, una interfaz de comunicaciones configurada para comunicarse con dicho servidor remoto, en el que dicho servidor remoto comprende una interfaz de comunicaciones configurada para comunicarse con dicho terminal de 45 servicio y una pluralidad de aplicaciones en el que cada una de dichas aplicaciones comprende un algoritmo definido de tal forma que sus estados, transiciones entre estados y condiciones pueden ser transmitidos mediante dicha interfaz de comunicaciones. El procedimiento comprende las etapas de: enviar una petición desde dicho terminal de servicio a dicho servidor remoto para descargar, al menos, un algoritmo de, al menos, una aplicación; transmitir dicho, al menos un algoritmo desde el servidor al terminal de servicio y cargarlo en dicha máquina de estados; generar un evento como 50 respuesta a una interacción de un usuario con uno de dichos módulos del terminal de servicio; asociar a dicho evento una información y enviarla a la máquina de estados; en la máquina de estados procesar dicha información y utilizando dicho, al menos un algoritmo ya cargado en la máquina de estados, obtener al menos una operación a realizar sobre alguno de los módulos del terminal de servicio; identificar el módulo sobre el que realizar dicha al menos una operación e interactuar con dicho módulo realizando dicha al menos una operación en el terminal de servicio.

55 Preferentemente, la información asociada a un evento comprende: un identificador del módulo que ha generado dicho evento y un identificador de operación que representa el evento particular generado por dicho módulo.

Esta información asociada a un evento comprende preferentemente además una pluralidad de parámetros que representan datos adicionales de información de dicho evento.

En una realización particular, cada uno de dichos módulos comprendidos en dicho terminal de servicio es un módulo periférico que comprende un módulo de control.

- 5 En una posible realización, la petición para descargar, al menos, un algoritmo de, al menos, una aplicación se envía cuando se enciende el terminal de servicio.

Alternativamente, la petición para descargar, al menos, un algoritmo de, al menos, una aplicación se envía en determinados momentos del día.

- 10 Alternativamente, la petición para descargar, al menos, un algoritmo de, al menos, una aplicación la realiza el usuario de forma manual a través del terminal de servicio.

Alternativamente, la petición para descargar, al menos, un algoritmo de, al menos, una aplicación la envía el terminal de servicio a través de un puerto de conexión también utilizado por el servidor remoto en un momento diferido en el tiempo.

Alternativamente, la petición para descargar, al menos, un algoritmo de, al menos, una aplicación se envía en el momento de ejecutar una operación en el terminal de servicio en la que sea necesaria dicha, al menos, una aplicación.

- 15 En otro aspecto de la invención se proporciona un sistema que comprende una pluralidad de terminales de servicio y un servidor remoto, donde la gestión de operaciones en cada terminal de servicio se lleva a cabo mediante el procedimiento descrito previamente.

- 20 En una realización particular, cada uno de dichos terminales de servicio comprende, además, un almacén de aplicaciones configurado para almacenar los algoritmos de las aplicaciones transmitidas desde el servidor remoto al terminal de servicio previamente a su carga en la máquina de estados.

Finalmente se proporciona un programa informático que comprende medios de código de programa informático adaptados para realizar las etapas del procedimiento descrito anteriormente, cuando dicho programa se ejecuta en un ordenador, un procesador de señal digital, una matriz de puertas programable en campo, un circuito integrado de aplicación específica, un microprocesador, un microcontrolador, y cualquier otra forma de hardware programable.

25 **Breve descripción de los dibujos**

Para complementar la descripción y con objeto de ayudar a una mejor comprensión de la invención, se proporciona un dibujo. Dicho dibujo forma parte integrante de la descripción e ilustra una realización de arquitectura preferida para la aplicación del procedimiento de la invención, que no debe interpretarse como que limita el ámbito de la invención, sino solo como un ejemplo de cómo se puede materializar la invención.

- 30 La figura 1 ilustra un esquema de la arquitectura de un punto de venta con tecnología STIP.

La figura 2 ilustra un esquema de la arquitectura de un terminal punto de venta y un servidor remoto, de acuerdo con una realización de la presente invención.

La figura 3 ilustra un esquema de la arquitectura de un terminal de punto de venta con almacén de aplicaciones y un servidor remoto de acuerdo con una realización de la presente invención.

35 **Descripción detallada de la invención**

La figura 2 ilustra un esquema de la arquitectura de una implementación de la invención. En concreto, en la figura 2 se representa un terminal de servicio o terminal punto de venta 201 y un servidor remoto 202.

- 40 De acuerdo con la Plataforma de Interoperabilidad de Terminales Pequeños (*Small Terminal Interoperability Platform*, STIP), es posible la representación de un terminal de servicio o terminal punto de venta (TPV) 201 como un conjunto de módulos 203, cada uno de ellos con un elemento de control y un elemento de gestión de los eventos. Ejemplos no limitativos de módulos 203 que pueden incluirse en un terminal TPV 201 son: lector de tarjetas sin contacto, diodos emisores de luz, lectores de cinta magnética, ranuras para lectura de tarjeta inteligente (del inglés, Smart Card Slot), teclado, impresora, interfaces de usuario, *beeper*, tarjeta de transporte, verificación de poseedor de tarjeta (del inglés, Card Holder Verification), comunicaciones, criptografía, fecha, fuente de alimentación, temporizador, fichero, http, servidor de http y XML.

- 45 Como puede observarse, el terminal de servicio o terminal punto de venta 201 comprende también una interfaz de comunicaciones 204 configurada para comunicarse con el servidor remoto 202, es decir, para gestionar las comunicaciones con el mismo y una máquina de estados 210 para controlar la ejecución del algoritmo de cada aplicación. La figura 2 muestra también un elemento de control 205, para interactuar y controlar los módulos 203 del TPV 201, y un elemento 206 para capturar los eventos que generan los módulos. Estos dos elementos 205 206 se asocian a cada uno de los módulos 203.
- 50

Por su parte, el servidor 202 alberga un elemento o interfaz 207 para gestionar las comunicaciones con el terminal de servicio o TPV 201. Además, el servidor 202 comprende la lógica 208 de todas las aplicaciones 209 que se quieran ejecutar. El servidor 202, por tanto, alberga distintas aplicaciones 209, en donde cada una de ellas contiene su propio algoritmo, permitiendo total flexibilidad.

- 5 El procedimiento de gestión de una operación en el terminal de servicio o terminal punto de venta 201 propuesto comprende los siguientes pasos:

En primer lugar, cuando se enciende el terminal de servicio o TPV 201 se envía una petición al servidor 202 para descargarse el algoritmo de la aplicación 209 que se vaya a ejecutar. En el servidor, los algoritmos de cada una de las aplicaciones 209 se encuentran definidos de manera formal, de tal forma que sus estados, transiciones entre estados y condiciones pueden ser representados y transmitidos al terminal de servicio 201. Seguidamente, el algoritmo de la aplicación 209 solicitada, adecuadamente formalizado es transmitido desde el servidor 202 al terminal de servicio 201 y posteriormente cargado en la máquina de estados 210. Una vez que el algoritmo está cargado en la máquina de estados 210, la aplicación 209 está disponible para ser utilizada. Cuando sucede algún tipo de interacción por parte de un usuario con alguno de los módulos 203 que componen el terminal de servicio o TPV 201, se genera un evento que es capturado por el elemento de gestión de los eventos 206. Por ejemplo, si se pulsa una tecla o si se pasa una tarjeta por el lector de tarjetas.

A continuación, el elemento de gestión de los eventos 206 caracteriza el evento, asociándole cierta información. Toda esta información es enviada a la máquina de estados 210.

La máquina de estados 210 recibe la información del evento generado en el terminal de servicio o TPV 201 y de acuerdo con el módulo 203, los datos adicionales y el estado de la aplicación 209, y en base al algoritmo de dicha aplicación 209, sus posibles estados, transiciones y condiciones, determina cuál es el siguiente paso que se debe realizarse en el terminal de servicio o TPV 201. Este paso se define en términos de una operación a realizar sobre alguno de los módulos 203 del terminal de servicio 201.

La máquina de estados 210 transmite al módulo de control 205 la operación a realizar, sobre qué módulo 203 y con qué parámetros. El módulo de control 205 se encarga de interactuar con el módulo 203 correspondiente, desencadenando la operación que se ha determinado en el algoritmo de la aplicación 209.

Este modelo permite que, haciendo uso recurrentemente de estos pasos, cualquier aplicación 209 pueda ser definida como un flujo de trabajo, que, con la condición de haber sido descargado del servidor 202, hace que dicho servidor sea el que determina la operación real.

El escenario descrito se basa en la ejecución de una sola aplicación 209 en el terminal de servicio 201. Dependiendo de la complejidad de las aplicaciones que se manejen puede interesar cargar varias aplicaciones 209 simultáneamente en el terminal de servicio o TPV 201. Para ello, como se indica en la figura 3, en otra realización de la presente invención, en la que los mismos elementos se referencian de forma similar a la figura 2 (201 301; 203 303, etc.) se incorpora un módulo adicional, denominado almacén de aplicaciones 311 en el terminal de servicio 301 que se encarga de la gestión de las aplicaciones 309, definidas en base a los algoritmos que representan.

En esta realización, al encender el TPV 301 los algoritmos de las aplicaciones 309 son cargados en el almacén de aplicaciones 311 en vez de en la máquina de estados directamente 310. Y en el momento en que se quiere ejecutar una determinada aplicación 309, la máquina de estados 310 carga su algoritmo desde el almacén de aplicaciones 311 permitiendo su ejecución de acuerdo a los pasos descritos anteriormente.

40 Realizaciones adicionales de la presente invención no permiten que el algoritmo o algoritmos de las aplicaciones 209 309 sean cargados en la máquina de estados 210 310 o en el almacén de aplicaciones 311 cuando el terminal de servicio 201 301 esté encendido, sino que esta carga se realice en otras situaciones determinadas, como por ejemplo:

- en determinados momentos del día, el terminal de servicio 201 301 solicita la carga de dichos algoritmos de las aplicaciones 209 309.
- al realizar operaciones en el TPV 201 301 en las que intervenga el servidor 202 302, marcar la respuesta para que el terminal de servicio 201 301 se encargue de solicitar la carga actualizada de la aplicación o aplicaciones 209 309.
- el usuario de forma manual realiza la petición de carga (bajo demanda) de la aplicación o aplicaciones 209 309.
- el terminal de servicio 201 301 a través de un puerto de conexión que es utilizado por el servidor 202 302 solicita la carga de la aplicación o aplicaciones 209 309, en un momento diferido en el tiempo.

50 En cualquiera de estos casos el terminal de servicio 201 301 almacena la lista y las versiones de las aplicaciones 209 309 que se encuentran ya instaladas y la envía al servidor 202 302 para minimizar el intercambio de información necesario, indicando al servidor 202 302 qué aplicación/es 209 309 hay que eliminar y qué aplicación/es 209 309 hay que actualizar.

55 En suma, se propone una arquitectura y un procedimiento que permiten total flexibilidad en la definición de aplicaciones que sean ejecutadas en un TPV 201 301, evitando el problema de tener que hacer modificaciones en el software del TPV

201 301.

La arquitectura propuesta es totalmente genérica, apoyada en la generación de eventos y elementos de control sobre los periféricos, lo que permite la incorporación a medida de los periféricos que sean precisos para cada escenario.

5 Además, al ser descargado desde el servidor 202 302 el algoritmo/s de la/s aplicación/es que se va/n a ejecutar, permite un control absoluto de la operativa que se ejecuta en el terminal TPV 201 301. Es de destacar que con esto se consigue una personalización total para cada terminal 201 301: se puede pensar en que cada terminal TPV 201 o grupo o terminales TPV pueden tener una configuración particular disponiendo de operaciones diferenciadas de las de otros terminales TPV 201 o grupos de TPV. La flexibilidad es total.

10 Otra ventaja, consecuencia de cargar los algoritmos de las aplicaciones 209 309 en el terminal de servicio 201 301 es que manteniendo la flexibilidad disminuyen las comunicaciones con el servidor 202 302, lo que conlleva menores volúmenes de datos intercambiados, y conduce al ahorro en tiempo y en costes imponiendo tiempos de respuesta menos limitados en las comunicaciones, permitiendo que sea una solución viable en entornos donde otras alternativas no lo serían, por el mayor número de comunicaciones con el servidor 201 301 que conllevan.

15

REIVINDICACIONES

- 1.- Un procedimiento para gestionar una operación en un terminal de servicio (201, 301) mediante un servidor remoto (202, 302), en el que dicho terminal de servicio (201, 301) comprende una pluralidad de módulos (203, 303), una máquina de estados (210, 310), configurada para cargar algoritmos correspondientes a aplicaciones y ejecutar dichas aplicaciones, y, una interfaz de comunicaciones (204, 304) configurada para comunicarse con dicho servidor remoto (202, 302), en el que dicho servidor remoto (202, 302) comprende una interfaz de comunicaciones (207, 307) configurada para comunicarse con dicho terminal de servicio (201, 301) y una pluralidad de aplicaciones (209, 309) en el que cada una de dichas aplicaciones (209, 309) comprende un algoritmo (208, 308) definido de tal forma que sus estados, transiciones entre estados y condiciones pueden ser transmitidos mediante dicha interfaz de comunicaciones (207, 307),
- estando el procedimiento **caracterizado por** las etapas de:
- enviar una petición desde dicho terminal de servicio (201, 301) a dicho servidor remoto (202, 302) para descargar, al menos, un algoritmo (208, 308) de, al menos, una aplicación (209, 309);
 - transmitir dicho, al menos un algoritmo (208, 308) desde dicho servidor (202, 302) al terminal de servicio (201, 301) y cargarlo en dicha máquina de estados (210, 310);
 - generar un evento como respuesta a una interacción de usuario con uno de dichos módulos (203, 303) del terminal de servicio (201, 301);
 - asociar a dicho evento una información y enviarla a la máquina de estados (210, 310);
 - procesar dicha información en la máquina de estados (210, 310) y obtener al menos una operación a realizar sobre cualquiera de los módulos (203, 303) del terminal de servicio (201, 301) utilizando dicho, al menos un algoritmo (208, 308) ya cargado en la máquina de estados (210, 310);
 - identificar el módulo (203, 303) sobre el que dicha al menos una operación está destinada a ser realizada e interactuar con dicho módulo (203, 303) realizando dicha al menos una operación en el terminal de servicio (201, 301).
2. El procedimiento de la reivindicación 1, en el que dicha información asociada a un evento comprende: un identificador del módulo (203, 303) que ha generado dicho evento y un identificador de operación que representa el evento particular generado por dicho módulo (203, 303).
3. El procedimiento de la reivindicación 2, en el que dicha información asociada comprende, además, una pluralidad de parámetros que representan datos adicionales de información de dicho evento.
4. El procedimiento de cualquiera de las reivindicaciones anteriores, en el que cada uno de dichos módulos (203, 303) comprendidos en dicho terminal de servicio (201, 301) es un módulo periférico que comprende un módulo de control (205, 305).
5. El procedimiento de cualquiera de las reivindicaciones anteriores en el que dicha petición para descargar, al menos, un algoritmo (208, 308) de, al menos, una aplicación (209, 309) es enviada cuando se enciende el terminal de servicio (201, 301).
6. El procedimiento de cualquiera de las reivindicaciones 1 a 4 en el que dicha petición para descargar, al menos, un algoritmo (208, 308) de, al menos, una aplicación (209, 309) es enviada en determinados momentos del día.
7. El procedimiento de cualquiera de las reivindicaciones 1 a 4 en el que dicha petición para descargar, al menos, un algoritmo (208, 308) de, al menos, una aplicación (209, 309) la realiza el usuario de forma manual mediante el terminal de servicio (201, 301).
8. El procedimiento de cualquiera de las reivindicaciones 1 a 4 en el que dicha petición para descargar, al menos, un algoritmo (208, 308) de, al menos, una aplicación (209, 309) es enviada por el terminal de servicio (201, 301) a través de un puerto de conexión también utilizado por el servidor remoto (202, 302) en un momento diferido en el tiempo.
9. El procedimiento de cualquiera de las reivindicaciones 1 a 4 en el que dicha petición para descargar, al menos, un algoritmo (208, 308) de, al menos, una aplicación (209, 309) es enviada en el momento de la ejecución de una operación en el terminal de servicio (201, 301) por lo que se necesita, al menos, una aplicación (209, 309).
10. Un sistema que comprende una pluralidad de terminales de servicio (201, 301) y un servidor remoto (202, 302), estando la pluralidad de terminales de servicio (201, 301) adaptada para realizar una gestión de las operaciones de acuerdo con los pasos del procedimiento de cualquier reivindicación anterior.

11. El sistema de la reivindicación 10, en el que cada uno de dichos terminales de servicio (301) comprende, además, un almacén de aplicaciones (311) configurado para almacenar los algoritmos (308) de las aplicaciones (309) transmitidos desde el servidor remoto (302) al terminal de servicio (301) previamente a la carga de dichos algoritmos (308) en la máquina de estados (310).

- 5 12. Un programa de ordenador que comprende medios de código de programa de ordenador adaptados para llevar a cabo el procedimiento según cualquiera de las reivindicaciones 1 a 9, cuando dicho programa se ejecuta en un ordenador, un procesador de señal digital, una matriz de puertas programable en campo, un circuito integrado de aplicación específica, un microprocesador, un microcontrolador, y cualquier otra forma de hardware programable.

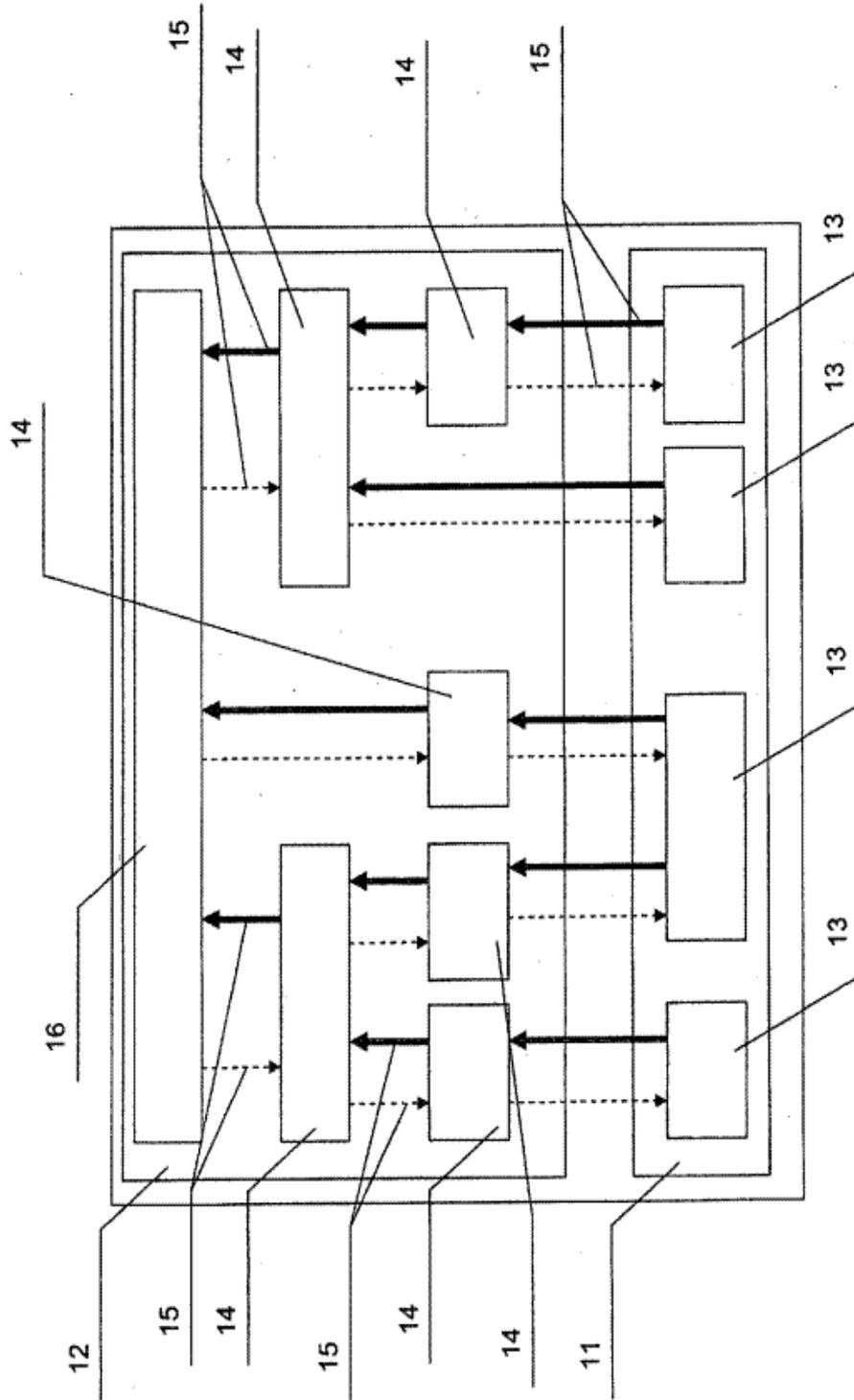


Fig. 1 (Técnica Anterior)

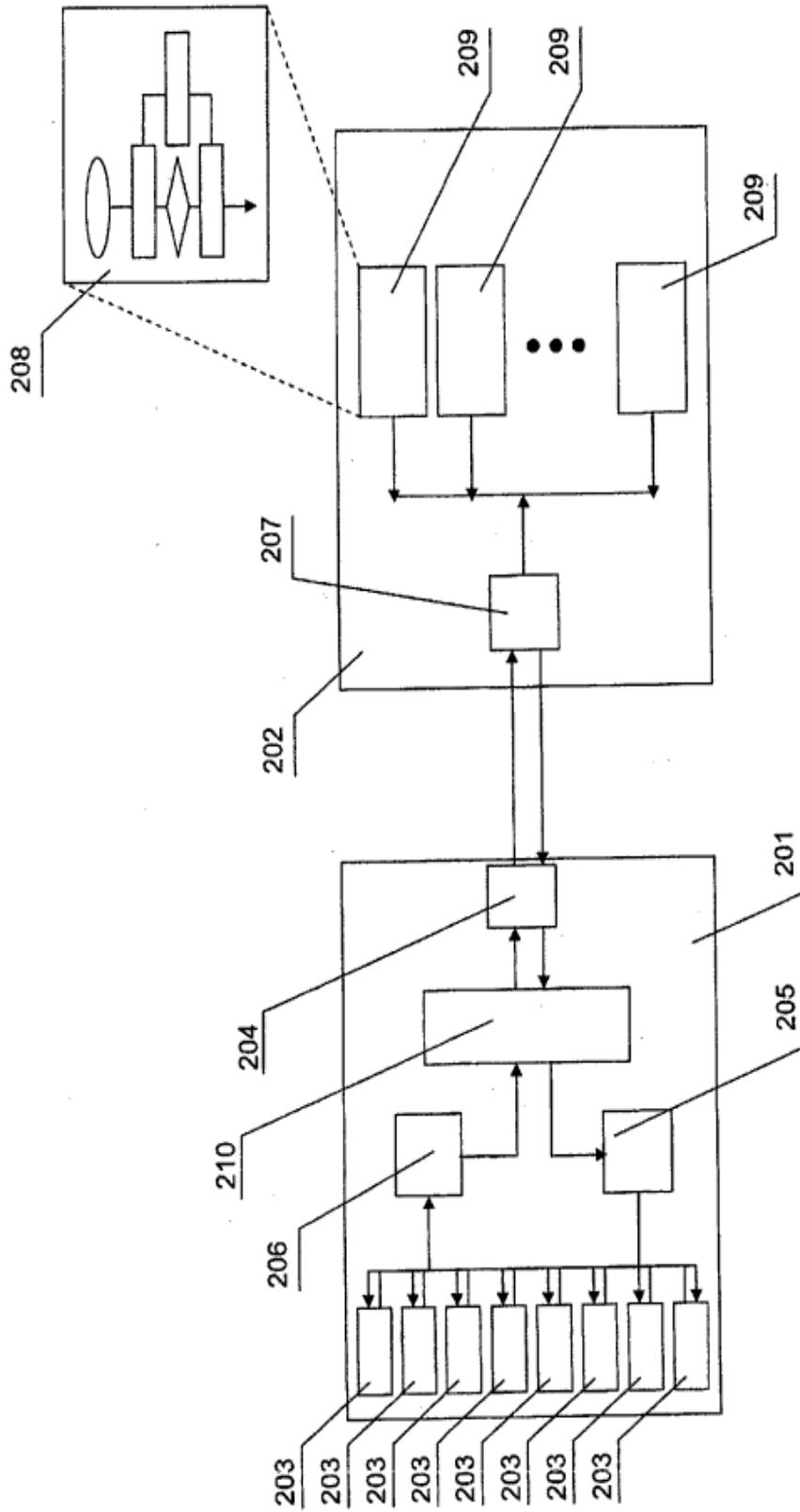


Fig. 2

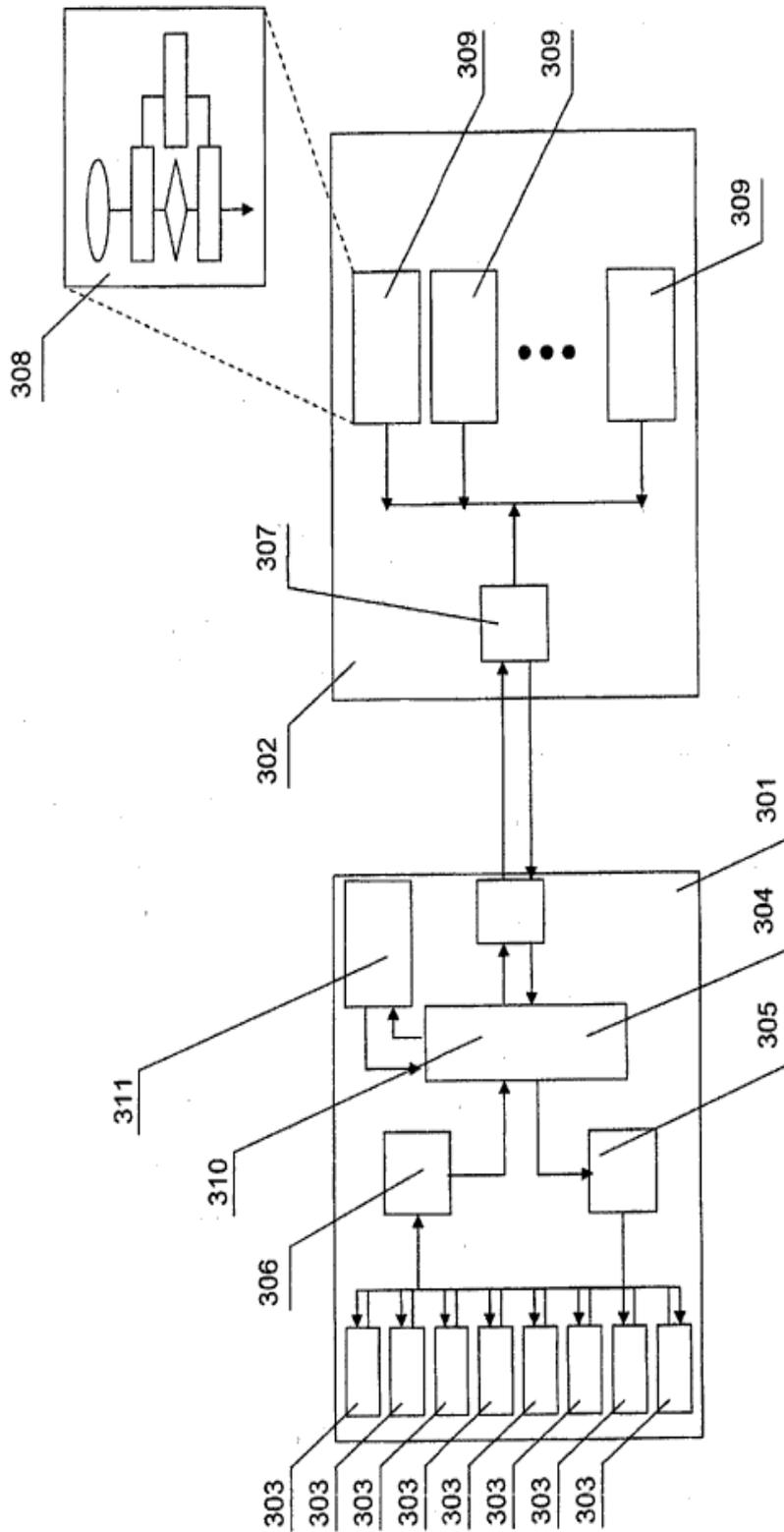


Fig. 3