

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 396 226**

51 Int. Cl.:

G06F 9/46 (2006.01)

G09B 5/00 (2006.01)

G09B 7/00 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **29.11.2001 E 01650143 (9)**

97 Fecha y número de publicación de la concesión europea: **21.11.2012 EP 1229470**

54 Título: **Seguimiento de datos relativos a cursos de aprendizaje por ordenador**

30 Prioridad:

30.11.2000 IE 20000973

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

20.02.2013

73 Titular/es:

**SKILLSOFT IRELAND LIMITED (100.0%)
Belfield Office Park, Clonskeagh
Dublin 4, IE**

72 Inventor/es:

**REDMOND, ANTHONY y
MANGAN, MARTIN**

74 Agente/Representante:

UNGRÍA LÓPEZ, Javier

ES 2 396 226 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

Seguimiento de datos relativos a cursos de aprendizaje por ordenador

5 **Introducción**

Campo de la invención

10 La invención se refiere a gestión de aprendizaje, y más en concreto a seguimiento del progreso de cursos de aprendizaje por ordenador.

Explicación de la técnica anterior

15 Nuestra Patente europea número 0690426B describe un sistema de aprendizaje por ordenador ("reproductor de curso") que administra y gestiona el contenido del curso para un estudiante. A medida que se ha desarrollado el aprendizaje por ordenador, muchos estudiantes utilizan varios cursos. Además, una organización concreta puede tener muchos estudiantes y muchos cursos, posiblemente alojados en una intranet. Esto da origen a la necesidad de hacer el seguimiento de los cursos según varios criterios incluyendo por estudiante, por grupo de organización, y por curso.

20 EP0809198 describe una disposición para acceso múltiple a base de datos para programas de aplicación.

25 US5859972 describe un depósito de servidor múltiple y ordenador cliente virtual de aplicación remota de servidor múltiple.

Así, la invención se refiere a proporcionar un sistema para el seguimiento efectivo y versátil de cursos de aprendizaje por ordenador.

30 **Resumen de la invención**

Según la invención, se facilita un sistema de seguimiento como se expone en la reivindicación 1.

35 En una realización, la interfaz común incluye un medio para consultar los módulos de traducción según un archivo preestablecido.

En otra realización, la interfaz común incluye un medio para recibir mensajes traducidos de los módulos de traducción.

40 En otra realización, el motor incluye un medio para gestionar una cola, y para establecer hilos para entrada y salida a la cola.

En una realización, el motor incluye un medio para asumir que un mensaje de actualización de datos es síncrono a no ser que el mensaje indique lo contrario.

45 En otra realización, el sistema incluye un medio de concentración, incluyendo un medio para crear un número de casos de cada SAO y para reutilizar los casos.

50 En otra realización, el medio de concentración incluye un gestor incluyendo un medio para gestionar una concentración de SAOs, para determinar un SAO libre, si lo hay, para poner un hilo solicitante en un estado dormido si un caso no está disponible temporalmente, y para ordenar que se cree un nuevo conjunto de casos SAO.

En una realización, el sistema incluye un medio para determinar durante la inicialización una cadena de conexión a pasar a un caso SAO para indicar la base de datos a abrir.

55 En otra realización, el gestor incluye un medio para reinicializar un caso SAO concentrado si opera incorrectamente.

En otra realización, el motor de seguimiento incluye un medio para mantener un hilo de entrada entre la interfaz común y la cola.

60 En una realización, el motor de seguimiento incluye un medio para mantener una pluralidad de hilos de lado de base de datos para dirigir mensajes desde la cola al medio de interfaz de base de datos.

65 En otra realización, cada hilo de lado de base de datos incluye un medio para esperar una respuesta del medio de interfaz de base de datos, y para dirigir directamente las respuestas recibidas a un módulo de traducción relevante para traducción y para recibir de nuevo las respuestas traducidas de los módulos de traducción.

En otra realización, cada hilo de lado de base de datos incluye un medio para dirigir directamente una respuesta traducida a un reproductor originante.

5 En una realización, el motor incluye un medio para mantener cada hilo de lado de base de datos en un estado activo o en un estado dormido.

En otra realización, el motor incluye un medio para conmutar un hilo de lado de base de datos a un estado activo en respuesta a la petición de hilo de entrada.

10 En una realización, el motor incluye un medio para escribir el contenido de la cola en un archivo de registro cuando se cierra inesperadamente, y para buscar automáticamente un archivo de registro al arranque.

En otra realización, la interfaz común incluye una función de tiempo agotado incluyendo un medio para terminar una conexión de reproductor a la expiración de un período de tiempo preestablecido.

15 En una realización, el medio de interfaz de base de datos incluye una función de tiempo agotado incluyendo un medio para terminar una conexión de base de datos de gestión de aprendizaje a la expiración de un período de tiempo preestablecido.

20 **Descripción detallada de la invención**

Breve descripción de los dibujos

25 La invención se entenderá más claramente por la descripción siguiente de algunas de sus realizaciones, dada a modo de ejemplo solamente con referencia a los dibujos acompañantes en los que:

La figura 1 es un diagrama de nivel alto que ilustra el contexto de un sistema de seguimiento de la invención.

30 Y la figura 2 es un diagrama que ilustra la arquitectura del sistema de seguimiento.

Descripción de las realizaciones

35 Con referencia a la figura 1, se utiliza un sistema de seguimiento 1 para el seguimiento del progreso de cursos que se ejecutan en navegadores de estudiante 2 que reciben el contenido mediante Internet 3 de servidores de reproductores de curso 4. En su otro lado, el sistema de seguimiento 1 escribe datos de progreso en bases de datos de gestión de aprendizaje 10.

40 En esta realización, todos los datos de curso son recibidos de servidores de reproductores de curso 4; sin embargo, pueden ser recibidos alternativamente de reproductores de curso autónomos que se ejecuten como aplicaciones en ordenadores de estudiante. Los reproductores y/o los servidores de reproductor que proporcionan los datos de curso y las bases de datos de gestión de aprendizaje pueden ser operados por terceras partes. Así, el operador del sistema de seguimiento 1 puede proporcionar un servicio de generación de datos de gestión de aprendizaje para cualquier reproductor o servidor de reproductor elegido para una tercera parte.

45 El sistema de seguimiento 1 opera con mínimo impacto en los reproductores y/o servidores de curso (en adelante "reproductores"). Una razón es que pueden comunicar asincrónicamente en un modo "dispara y olvida". Esto es posible mediante el uso de una capa de comunicación de nivel bajo (TCP) que asegure la recepción segura.

50 El sistema de seguimiento 1 puede manejar la captura de datos de reproductores a cualquier frecuencia para adaptarse al reproductor, siendo el rango de segundos a horas. Algunos reproductores pueden estar configurados para la actualización, en base al tiempo, cada dos segundos, mientras que otros pueden estar configurados para actualización solamente después del final de un curso. En el último caso, el reproductor realiza seguimiento interno durante el curso.

55 Con referencia a la figura 2, el sistema de seguimiento 1 incluye un motor de seguimiento central 20. Una interfaz común 21 comunica con todos los reproductores, y dirige señales a un módulo de traducción elegido 22 ("traductor"). El motor 20 gestiona hilos entre los traductores 22 y las interfaces de base de datos 24. Hay una interfaz de base de datos 24 para cada base de datos de aprendizaje 10. El motor 20 realiza puesta en cola de modo que haya puesta en memoria intermedia efectiva entre los reproductores 4 y las bases de datos 10.

60 Con más detalle, el sistema de seguimiento 1 también pone en interfaz en el lado de reproductor con sistemas de cliente que piden datos, tal como un motor de prueba o una herramienta informante. Los módulos de traducción 22 traducen a un lenguaje común usado por el motor de seguimiento 2, CMIML2. Si los datos entrantes están en este formato, son dirigidos directamente al motor 20 sin traducción.

65 El motor 20 pone en cola peticiones/mensajes de datos recibidos y los analiza mientras están en las colas. A

continuación realiza la operación de base de datos mediante las interfaces 24, escribe o lee. Donde el mensaje entrante es síncrono, el motor dirige una respuesta de vuelta a la aplicación solicitante.

5 Cada interfaz de base de datos 24 es un SAO (Objeto de Acceso a Esquema). Éste es un objeto que entiende cómo la información de seguimiento de estudiante es almacenada en un tipo concreto de base de datos de gestión de aprendizaje (por ejemplo, la base de datos de seguimiento de estudiante en Smartforce Campus™). Cada SAO tiene la misma interfaz expuesta. Cada SAO se define en un módulo, que tiene un método de factoría definido para crear un ejemplo del SAO. Se define la interfaz y el método de factoría que debe existir en el módulo de SAO. Un SAO puede ser suministrado como un plug-in modular al sistema 1, típicamente por los proveedores de las bases de datos 10.

15 Para crear un ejemplo de un SAO, el sistema 1 carga el módulo requerido (localizado por su nombre y vendedor), e invoca el método de factoría definido para obtener un ejemplo de un objeto que es conforme con la interfaz SAO conocida.

20 Hay una anulación para este proceso. Es posible que el administrador de preparación prepare el sistema 1 en una forma que especifique "Todas las peticiones que vengan del reproductor X deberán ser tratadas por el SAO Y". Esto se logra cambiando el archivo "redirect.dat". Éste es un archivo, que el administrador puede editar, que redirige todas las peticiones para el SAO A a peticiones para el SAO B. Este archivo "redirect.dat" es leído cuando el sistema 1 arranca, y permite que los sistemas de cliente comuniquen con bases de datos de seguimiento de la opción del administrador, aunque la combinación concreta de sistema de cliente y base de datos 10 nunca fuese diseñada para trabajar conjuntamente. Lo que sigue es un archivo redirect.dat muestra.

```

; Redirect.dat
; SAO Redirection
;
; This file is a sample, to illustrate SAO redirection.
;
;
; The entries in this file are processed in the order in which
; they occur.
;
; The first entry (below) maps all requests for the
2.0|cbt|WebPlus SAO to the
1.0|VendorA|Learning_Management_System_A SAO.
;
2.0|cbt|WebPlus -> 1.0|VendorA|Learning_Management_System_A
;
;
; The next entry maps all other 2.0|cbt SAO requests to the
2.0|cbt|campus SAO.
; Note that the asterix '*' is a wildcard, and can correspond to
any name.
;

2.0|cbt|* -> 2.0|cbt|campus
;
;
; The last entry is a "catch-all". It maps all other requests to
the 2.0|cbt|null SAO.
;
;
*|*|* -> 2.0|cbt|null

```

El sistema 1 aúna SAOs. Esto permite crear un número de casos de cada SAO, y reutilizar estos casos para cada petición. Cuando se recibe una petición de un cliente, la petición es examinada. La petición contiene dos campos que indican qué producto (y de qué vendedor) ha enviado la petición. Esto se correlaciona directamente con un SAO. El sistema 1 consulta un “gestor de SAO” para comprobar si este SAO ya está en memoria. Esto da lugar a una de dos situaciones:

1. Si EL SAO requerido ya está en memoria, el gestor de SAO examina su grupo de SAOs para hallar un caso libre del tipo de SAO requerido. Si no hay ninguno disponible, el hilo se pone en un estado “dormido”, a la espera de que un SAO esté disponible.

2. Si EL SAO requerido no está en memoria, el gestor de SAO localiza el módulo correcto por nombre, lo carga, e invoca el método de factoría del módulo. Invoca el método de factoría múltiples veces, recuperando cada vez un caso nuevo del SAO. Estos casos se ponen en un grupo para uso futuro.

Cada módulo SAO debe tener un archivo INI correspondiente (que el administrador puede editar), conteniendo (al menos) el número de casos de DICHO SAO que deberán ser agrupados por el sistema 1, y la cadena de conexión que se debe pasar al caso del SAO con el fin de que el SAO sepa qué base de datos abrir. Esto permite al administrador apuntar el SAO a un caso diferente de una base de datos, sin cambiar el código. El sistema 1 examina este archivo INI antes de crear el grupo de SAOs, y usa el valor “Tamaño de grupo” para decidir cuántos casos del SAO crear, y pasa la cadena de conexión al SAO al crear el SAO.

Si, en alguna etapa, el sistema 1 determina que un caso de un SAO deja de operar correctamente, intenta reinicializar dicho caso del SAO. Lo hace desinicializando el SAO, intentando luego reinicializarlo de nuevo. Esto permite al sistema 1 recuperarse automáticamente, si la base de datos se cayese o tuviese algún otro fallo.

El motor de seguimiento 20 existe como un proceso que se ejecuta en un servidor. Cada módulo de traducción 22 es un objeto, almacenado en un módulo (un DLL en Windows, un JavaBean en Java). Hay un parámetro (en un archivo “sfconnect.ini”) que el sistema 1 comprueba al arrancar, indicando qué traductores 22 se han de usar. Los traductores requeridos 22 son cargados y usados mientras el sistema 1 se está ejecutando. Cuando se recibe una petición de un cliente, la petición se pasa (usando una llamada de método) a cada traductor 22 por turno. El primer traductor que devuelve indicando que puede manejar la petición es elegido para manejar dicha petición. El orden en que los traductores son consultados es según una lista definida por el usuario, almacenada en un archivo XML y analizada al arrancar.

Así, la interfaz común 21 opera como un cliente de los traductores 22, llamándolos cuando sean necesarios. Un hilo de entrada 30 del motor de seguimiento 20 distribuye mensajes entrantes de la interfaz común 21 a una cola 31. El hilo 31 también activa hilos de lado de base de datos 32 cuando son requeridos.

Los mensajes puestos en la cola 31 incluyen señalizadores que indican el reproductor o cliente solicitante. Además, la interfaz común 21 inserta un señalizador que indica qué traductor 22 se usó, si se usó alguno.

La cola 31 es una estructura de datos a prueba de hilos en memoria, que aumenta y se contrae cuando es necesario. Los parámetros de administración especifican el número máximo permitido de peticiones que pueden estar en cola en cualquier tiempo.

Los hilos de lado de base de datos 32 son hilos de ejecución permanente que operan, cada uno, solamente en un mensaje a la vez. Tienen estados activo y dormido, conmutando del estado dormido al activo en respuesta al hilo de entrada 30.

En el estado activo, un hilo 32 lee el mensaje actualmente en la cabecera de la cola. Hace una llamada (invocación) en el SAO relevante 24 y espera una respuesta si el mensaje es síncrono. Presume que un mensaje es síncrono a no ser que un señalizador en el mensaje indique lo contrario.

A la recepción de la respuesta del SAO 24, el hilo 32 determina si se precisa traducción. Si es así, hace una llamada al traductor relevante 22 (indicado en el mensaje propiamente dicho). De nuevo, espera la respuesta del traductor. La respuesta es dirigida entonces directamente a la toma de red para transmisión directa al reproductor o cliente originante.

La integridad de los enlaces de comunicación es asistida por programas de intervalo de espera que terminan conexiones entre el reproductor o cliente y la interfaz común 21 y entre los SAOs y las bases de datos 10 si transcurren los tiempos preestablecidos. Esto asegura que UN SAO con fallo no afecte a la operación general del sistema.

Si el sistema 1 detecta un problema grave o recibe un mensaje del administrador indicando que deberá ser descargado de memoria, se debe parar. Antes de pararse, examina su cola. Las peticiones presentes en la cola son

vaciadas a un archivo de registro justo antes de la parada. Cada vez que arranca, busca este archivo de registro. Si existe, el contenido del registro es puesto en cola de nuevo para procesado, y se borra el archivo de registro. Esto ayuda a asegurar que no se pierdan peticiones, aunque el sistema 1 se deba parar.

5 El sistema 1 puede operar en un modo asíncrono de “dispara y olvida”. Esto permite a un cliente enviar una petición (típicamente actualización), y no esperar una respuesta. El cliente puede confiar en que la petición será procesada porque:

10 1. El protocolo de red subyacente, TCP, garantiza que cualesquiera datos son enviados satisfactoriamente solamente cuando el destino haya recibido los datos.

2. Cuando el sistema 1 recibe los datos, su mecanismo de puesta en cola garantiza que serán manejados, aunque mientras tanto tenga lugar una catástrofe.

15 El sistema 1 puede usar una red de dos formas.

1. Puerto dedicado

20 En este modo, usa un puerto dedicado, y comunica directamente usando el lenguaje de opción (por ejemplo CMIML) por la red/internet en dicho puerto.

2. Soporte HTTP

25 En este modo, se permiten peticiones como en (1) anterior. Sin embargo, el sistema 1 también permite que los clientes “lleven” peticiones por HTTP, usando el mismo puerto. Por ejemplo, el administrador podría configurar el sistema 1 para operar en el puerto 80, y activar el soporte HTTP. Esto permite a las aplicaciones de cliente enviar las peticiones directamente al sistema 1 por el puerto 80 o alternativamente encapsular las peticiones (y respuestas) en peticiones HTTP en el puerto 80. Esto permite que la comunicación con el sistema 1 tenga lugar por Internet incluso donde haya cortafuegos. (Los cortafuegos permiten típicamente que pasen comunicaciones HTTP, pero
30 bloquean otros muchos tipos de comunicación).

Los parámetros que indican al sistema 1 qué puerto usar, y si permitir o no que las peticiones sean llevadas en HTTP, están situados en el archivo SFCONNECT.INI. El administrador puede cambiar estos parámetros.

35 Se apreciará que la invención permite una actualización e información sumamente versátiles a partir de bases de datos de gestión de aprendizaje. Los traductores, los hilos y los SAOs son especialmente ventajosos para la versatilidad y también para la robustez.

REIVINDICACIONES

- 5 1. Un sistema de seguimiento (1) incluyendo un medio para recibir datos de actualización de reproductor de curso, y para escribir los datos de actualización en una base de datos de gestión de aprendizaje, **caracterizado** porque
- 10 el sistema incluye además una pluralidad de módulos de traducción (22), incluyendo cada uno un medio para traducir de un lenguaje de reproductor a un lenguaje de motor común, y viceversa;
- 15 cada módulo de traducción es un objeto instanciado al arrancar;
- 20 el sistema (1) incluye además una interfaz común (21) incluyendo un medio para la interfaz con todos los reproductores (4), y para consulta dirigiendo un mensaje recibido a todos los módulos de traducción (22); donde cada módulo de traducción (22) incluye un medio para analizar mensajes recibidos y, si puede traducir el mensaje, indicarlo como tal; y donde la interfaz común (21) incluye un medio para activar un módulo de traducción (22) que responde positivamente;
- 25 el sistema incluye un motor de seguimiento (20) incluyendo un medio para gestionar la comunicación unidireccional para actualizaciones asíncronas de datos de reproductor de curso, y para gestionar la comunicación bidireccional para actualizaciones síncronas de datos de reproductor de curso y respuestas;
- 30 el sistema (1) incluye un medio de interfaz de base de datos para comunicación (24) con una pluralidad de bases de datos de gestión de aprendizaje (10);
- 35 donde el medio de interfaz de base de datos (24) incluye un Objeto de Acceso a Esquema (SAO) asociado con cada base de datos de gestión de aprendizaje (10), donde todos los objetos de acceso a esquema (24) tienen la misma interfaz expuesta al motor de seguimiento (20), y donde el sistema (1) incluye un medio tanto para la preparación previa como para la posterior modificación de asociaciones entre reproductores y objetos de acceso a esquema ;
- 40 donde el motor (20) incluye un medio para gestionar una cola (31), y para establecer hilos para entrada y salida a la cola;
- 45 donde el motor de seguimiento (20) incluye un medio para mantener una pluralidad de hilos de lado de base de datos (32) para dirigir mensajes de la cola (31) al medio de interfaz de base de datos (24);
- 50 donde cada hilo de lado de base de datos (32) incluye un medio para esperar una respuesta del medio de interfaz de base de datos (24), y para dirigir directamente las respuestas recibidas a un módulo de traducción relevante (22) para traducción y para recibir de nuevo las respuestas traducidas de los módulos de traducción; y
- 55 donde cada hilo de lado de base de datos (32) incluye un medio para dirigir directamente una respuesta traducida a un reproductor originante.
- 60 2. Un sistema según la reivindicación 1, donde la interfaz común (21) incluye un medio para consultar los módulos de traducción (22) según un archivo preestablecido.
- 65 3. Un sistema según las reivindicaciones 1 o 2, donde la interfaz común incluye un medio para recibir mensajes traducidos de los módulos de traducción.
4. Un sistema según la reivindicación 1, donde el motor (20) incluye un medio para suponer que un mensaje de actualización de datos es síncrono a no ser que el mensaje indique lo contrario.
5. Un sistema según cualquier reivindicación precedente, donde el sistema (1) incluye un medio de concentración, incluyendo un medio para crear un número de casos de cada SAO y para reutilizar los casos.
6. Un sistema según la reivindicación 5, donde el medio de concentración incluye un gestor incluyendo un medio para gestionar una concentración de SAOs, para determinar un SAO libre si existe, para poner un hilo solicitante en un estado dormido si un ejemplo no está temporalmente disponible, y para ordenar que se cree un nuevo conjunto de casos SAO.
7. Un sistema según la reivindicación 6, donde el sistema (1) incluye un medio para determinar durante la inicialización una cadena de conexión a pasar a un caso SAO para indicar la base de datos (10) a abrir.
8. Un sistema según las reivindicaciones 6 o 7, donde el gestor incluye un medio para reinicializar un caso SAO concentrado si opera incorrectamente.
9. Un sistema según cualquiera de las reivindicaciones 1 a 8, donde el motor de seguimiento (20) incluye un medio para mantener un hilo de entrada (30) entre la interfaz común (21) y la cola (31).

10. Un sistema según cualquier reivindicación 1, donde el motor (20) incluye un medio para mantener cada hilo de lado de base de datos (32) en un estado activo o un estado dormido.
- 5 11. Un sistema según la reivindicación 1, donde el motor (20) incluye un medio para conmutar un hilo de lado de base de datos (32) a un estado activo en respuesta a la petición de hilo de entrada.
12. Un sistema según cualquier reivindicación precedente, donde el motor (20) incluye un medio para escribir el contenido de la cola en un archivo de registro cuando se para inesperadamente, y para buscar automáticamente un
10 archivo de registro al arrancar.
13. Un sistema según cualquier reivindicación precedente, donde la interfaz común (21) incluye una función de tiempo agotado incluyendo un medio para terminar una conexión de reproductor a la expiración de un período de
15 tiempo preestablecido.
14. Un sistema según cualquier reivindicación precedente, donde el medio de interfaz de base de datos incluye una función de tiempo agotado incluyendo un medio para terminar una conexión de base de datos de gestión de aprendizaje (10) a la expiración de un período de tiempo preestablecido.
- 20 15. Un producto de programa de ordenador incluyendo código de software para completar un sistema según cualquier reivindicación precedente al ejecutarse en un ordenador digital.

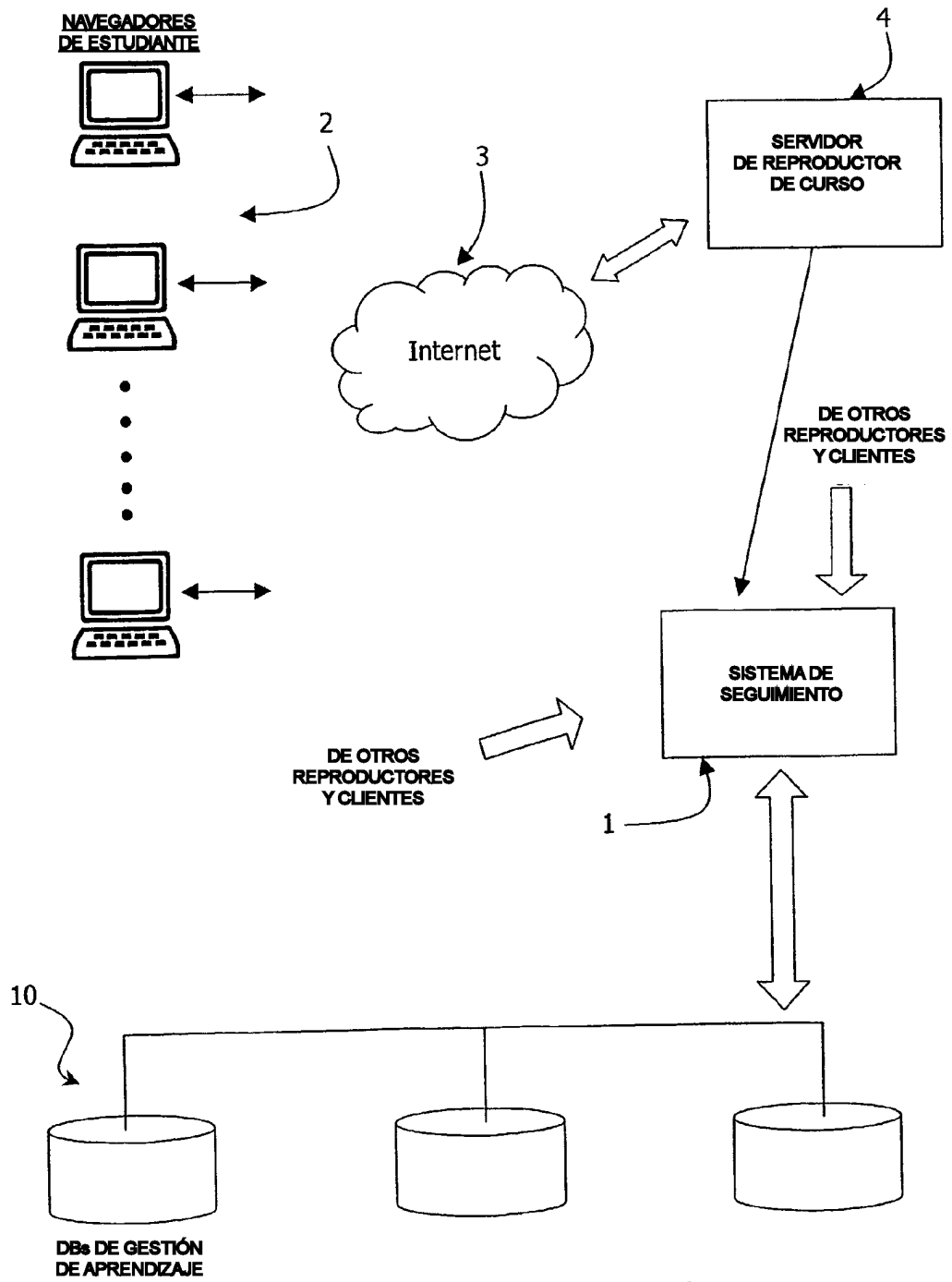


Fig. 1

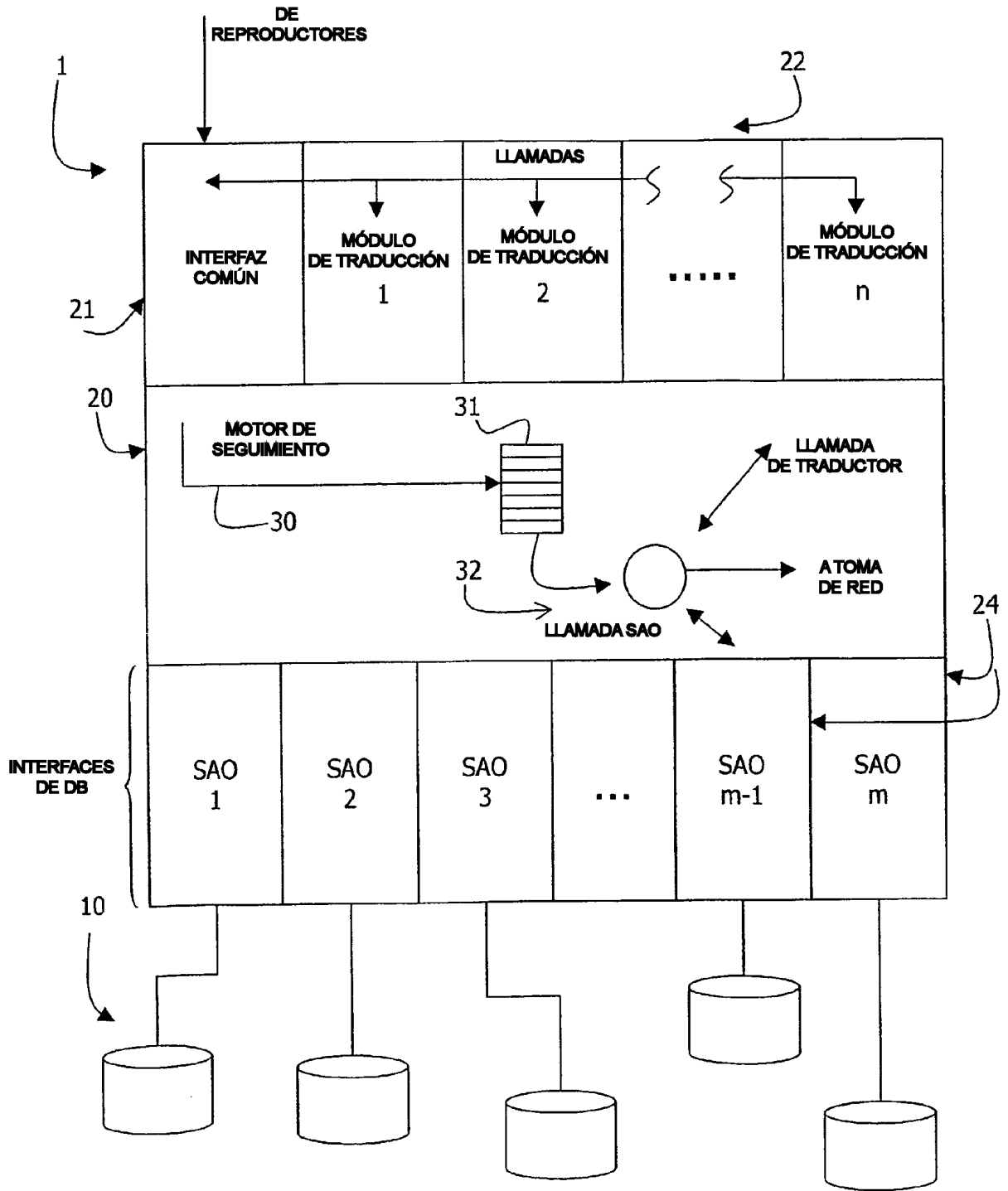


Fig. 2