

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 796 305**

51 Int. Cl.:

G06F 9/50 (2006.01)

G06Q 10/06 (2012.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

96 Fecha de presentación y número de la solicitud europea: **31.08.2016 E 16186721 (3)**

97 Fecha y número de publicación de la concesión europea: **22.04.2020 EP 3139273**

54 Título: **Método para determinar los trayectos paralelos de proceso en los datos del proceso**

30 Prioridad:

04.09.2015 US 201514846461

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

26.11.2020

73 Titular/es:

**CELONIS SE (100.0%)
Theresienstrasse 6
80333 München, DE**

72 Inventor/es:

**KLENK, MARTIN;
RINKE, ALEXANDER y
NOMINACHER, BASTIAN**

74 Agente/Representante:

SÁNCHEZ SILVA, Jesús Eladio

ES 2 796 305 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

Método para determinar los trayectos paralelos de proceso en los datos del proceso

5 Campo de la invención

La invención se refiere con un método flexible para procesar, reconstruir y analizar procesos comerciales reales sobre la base de datos de eventos generados por instancias del proceso. El objetivo es poder observar, analizar y visualizar un estado real de uno o varios procesos.

10

Estado de la técnica y antecedentes de la invención

Es bien conocido el análisis de procesos comerciales sobre la base de los rastros digitales en los sistemas informáticos. La totalidad de las etapas del proceso almacenadas en un sistema informático da como resultado un proceso que se puede reconstruir. Por ejemplo, se pueden visualizar las transacciones de los sistemas de planificación de los recursos institucionales.

15

El análisis de los datos no tiene lugar en los sistemas fuente (por ejemplo, el sistema de planificación de recursos empresariales), sino en una base de datos separada de los sistemas fuente. De acuerdo con el llamado proceso ETL, los datos se extraen de los sistemas fuente, los datos extraídos se transforman y los datos transformados se almacenan en un banco de datos. El análisis de los procesos se realiza entonces en el banco de datos almacenado en esta base de datos. Por ejemplo, W. Van Der Aalst: "Process Mining Discovery, Conformance and Enhancement of Business Processes" (2011) describe un flujo de trabajo para obtener los resultados de la minería de procesos a partir de fuentes de datos heterogéneas.

20

25

Una desventaja de los métodos conocidos en el estado de la técnica es, por ejemplo, que las opciones de análisis basadas en los datos generados por las instancias del proceso real (protocolos de instancias del proceso) son muy limitadas en cuanto a flexibilidad, desempeño y capacidad de analizar procesos de cualquier complejidad. Solo se pueden realizar análisis fijos predefinidos. El análisis de procesos paralelos muy complejos es poco preciso y el desempeño de los algoritmos no es suficiente para procesar conjuntos de datos muy grandes con varios cientos de millones de instancias.

30

Objetivo de la invención

El objetivo de la invención es proporcionar soluciones que permitan un análisis más simple, mejorado, flexible y eficiente de procesos potencialmente paralelos de cualquier complejidad.

35

Solución de acuerdo con la invención

De acuerdo con la invención, este objetivo se logra mediante un método de acuerdo con las reivindicaciones independientes. Las reivindicaciones dependientes contienen modalidades ventajosas y mejoras de la invención.

40

Breve descripción de las figuras

Los detalles y características de la invención, así como sus modalidades concretas, resultan de la siguiente descripción en relación con el dibujo. Se muestra:

45

En la Figura 1, un diagrama de bloques de un sistema de acuerdo con la invención;

En la Figura 2, una estructura de datos de un protocolo del proceso de acuerdo con la invención;

En la Figura 2 (a), un ejemplo de protocolo del proceso particionado;

En la Figura 3, datos que explican la determinación de los subprocesos paralelos basados en el protocolo del proceso;

50

En la Figura 4, un diagrama de bloques con un motor APE de acuerdo con la invención; y

En la Figura 5, un ejemplo de un proceso visualizado.

Descripción detallada de la invención

55

La Figura 1 muestra un diagrama de bloques de un sistema de acuerdo con la invención con el que los procesos reales (por ejemplo, los procesos comerciales) se pueden supervisar y, si fuera necesario, visualizar ad hoc. También se pueden registrar ad hoc los cambios en los procesos.

60

A continuación se explica con más detalle el análisis de los protocolos del proceso.

Si los procesos reales, como los procesos comerciales, se ejecutan con la ayuda de uno o más sistemas informáticos, las instancias individuales del proceso dejan rastros de datos en los sistemas informáticos. Por ejemplo, un proceso de producción puede crear múltiples rastros de datos en diferentes sistemas informáticos (cada producto producido, es decir, cada instancia del proceso deja rastros en diferentes máquinas del sistema de gestión de inventario, etc.). En este ejemplo, si observamos los rastros que deja una ejecución completa del proceso de producción (como un producto acabado) en los sistemas informáticos, estos rastros contienen mucha información sobre el trayecto real del proceso. Se generan

65

5 marcas de tiempo en cada etapa de la producción. Esto permite, por ejemplo, rastrear qué etapa del proceso se ejecutó en qué momento. De esto se puede deducir, entre otras cosas, qué etapas del proceso se ejecutaron con qué frecuencia y en qué orden, y cuánto tiempo transcurrió entre las etapas de producción. Si observamos ahora un gran número de ejecuciones del proceso de producción y la totalidad de los datos que cada ejecución ha dejado en cada etapa del proceso, obtenemos el protocolo del proceso. En el caso de los procesos con muchas etapas del proceso, que se ejecutan en diferentes sistemas informáticos, este protocolo del proceso puede ser muy extenso y contener un gran número de protocolos de datos por instancia del proceso (es decir, para cada ejecución del proceso). A partir del protocolo del proceso, se pueden realizar ahora varias consultas sobre el flujo de procesamiento real. Por ejemplo, se puede consultar 10 cuán homogéneo es el proceso de producción (es decir, en qué medida difieren entre sí las distintas instancias del proceso de producción), cómo es el proceso principal y qué patrones suelen dar lugar a desviaciones del proceso principal.

Por consiguiente, un protocolo del proceso se refiere a los datos generados por las instancias de un proceso real y se distribuye en uno o más sistemas informáticos.

15 El protocolo del proceso se puede fusionar y almacenar en un dispositivo de almacenamiento en el servidor. Alternativamente, el protocolo del proceso también se puede almacenar en un dispositivo de almacenamiento que esté físicamente separado del servidor. En ambos casos resulta ventajoso, de acuerdo con la invención, que el protocolo del proceso se almacene en la memoria. Además, es ventajoso que el protocolo del proceso se almacene en la memoria de trabajo en columnas.

20 El objetivo de la invención es responder a consultas arbitrarias al protocolo del proceso para proporcionar análisis muy flexibles de los procesos reales de la empresa.

25 El sistema de acuerdo con la invención comprende un dispositivo servidor (Server) que se puede acoplar con un número de clientes. El dispositivo servidor se adecua aún más para acceder a un protocolo del proceso.

Los clientes se adecuan para hacer consultas al servidor y recibir respuestas del mismo. Los datos recibidos del cliente se pueden preparar y visualizar en un dispositivo de visualización.

30 En una modalidad ventajosa de la invención, los clientes se adecuan para hacer consultas al servidor en una Notación APE (consultas APE) (APE = Ejecución Avanzada de Álgebra de Procesos). La Ejecución Avanzada de Álgebra de Procesos APE es un lenguaje de consulta proporcionado por la invención.

La APE se explica más detalladamente a continuación.

35 El protocolo del proceso tiene la propiedad de que para cada instancia del proceso existe una cantidad indefinida de protocolos de datos (potencialmente estructurados de manera diferente) de las distintas etapas del proceso. Esto significa que puede haber un número diferente de protocolos de datos para cada instancia (en dependencia del número de etapas del proceso por las que haya pasado la instancia y de la frecuencia).

40 La APE se adapta para consultar el protocolo del proceso de manera particularmente fácil y eficiente, por ejemplo, para filtrar, comparar o reconstruir los procesos almacenados en el protocolo del proceso. Además, la APE ofrece la posibilidad de definir con flexibilidad diferentes consultas y de combinar las ventajas de los llamados operadores del proceso y las funciones convencionales de las bases de datos de manera flexible.

45 Las funciones convencionales de las bases de datos (también llamadas operadores de bases de datos) pueden incluir: Funciones de agregación (promedio, suma, etc.), a menudo en combinación con la agrupación; funciones analíticas (mediana, cuantiles, etc.); funciones de conversión (cadena hasta la fecha, etc.); funciones criptográficas (calcular el valor hash, cifrar y descifrar, etc.); funciones de fecha (calcular las diferencias de tiempo, etc.); funciones lógicas; funciones matemáticas; funciones de cadena (izquierda, subcadena, etc.); operaciones de unión (vinculación de varias tablas).

50 De acuerdo con la invención, para el análisis de los procesos se accede directamente al protocolo del proceso. Gracias a la estructura de datos del protocolo del proceso de acuerdo con la invención (vea la descripción de la Figura 2) y al lenguaje de consulta APE de acuerdo con la invención, se pueden realizar con eficiencia los más diversos análisis ad hoc de los procesos, incluso con cantidades de datos muy grandes.

55 Un operador del proceso es un operador o función que se aplica a los procesos reconstruidos a partir del protocolo del proceso y proporciona como resultado un protocolo, tabla, valor o gráfico del proceso.

60 Ejemplos de operadores del proceso son:

- Coincidir (determinar las instancias del proceso que coinciden con un determinado patrón);
- Reprocesar (por ejemplo, determinar la frecuencia de ciertas actividades en los procesos);
- Indicador de bucles (por ejemplo, determinar la frecuencia de los bucles en los procesos);
- Determinar los gráficos del proceso a partir de una serie de instancias del proceso;
- 65 • Restar los gráficos del proceso (por ejemplo, determinar una diferencia entre dos gráficos del proceso);

- Determinar los tiempos de ejecución (por ejemplo, los tiempos de ejecución entre dos etapas del proceso predeterminados en una serie de instancias del proceso);
- Extraer los subprocessos; y/o
- Dividir los gráficos del proceso (por ejemplo, en base a criterios predeterminados).
- Agrupar, (por ejemplo, elaboración de un gráfico del proceso después de resumir ciertas etapas del proceso)

Primero hay que definir formalmente el lenguaje de consulta APE (Ejecución Avanzada de Algebra de Procesos): Los operadores del proceso soportados por la APE implementan una ampliación del álgebra relacional. La base del álgebra relacional, que en esta descripción se asume que ya se conoce, es

$$\text{relación } R = \{(r_{1,1}, r_{1,2}, \dots, r_{1,m}), \dots, (r_{n,1}, r_{n,2}, \dots, r_{n,m})\}$$

$$(r_{1,1}, \dots, r_{1,m}) \dots (r_{n,1}, r_{n,2}, \dots, r_{n,m}) \text{ tupla}$$

$$r_{1,1}, \dots, r_{n,m} \text{ atributos}$$

Un álgebra relacional requiere las siguientes operaciones como sistema mínimo:

- Proyección
- Selección
- Producto vectorial
- Asociación
- Diferencia y renombramiento

El álgebra relacional es la base de los sistemas de bases de datos relacionales y los correspondientes lenguajes de consulta (como el SQL). Debido a que el protocolo del proceso y los operadores del proceso solo pueden ser representados en un álgebra relacional con grandes restricciones, esta álgebra se amplía de acuerdo con la invención para la APE.

A continuación se describe el protocolo del proceso:

Asumamos

$$R = \{(r_{1,1}, r_{1,2}, \dots), \dots, (r_{n,1}, r_{n,2}, \dots)\}$$

una relación.

Un protocolo del proceso se define entonces como una relación de proceso

$$\{E_1, E_2, \dots, E_k\}, E_i \subset R$$

$$\forall i, j | i \neq j : E_i \cap E_j = \emptyset$$

$$\bigcup_i E_i = R$$

Además, existe una relación de orden para cada protocolo de subprocesso:

$$O_i = (E_i, Q_i)$$

con

O_i...la cantidad ordenada, así como

Q_i...relación de orden sobre E_i.

Así pues, el protocolo del proceso se define como

$$E := \bigcup_i O_i$$

Las cantidades E_i también se denominan caso o instancia del proceso.

Las instancias del proceso son en sí mismas relaciones y como tales son compatibles con todos los operadores relacionales.

Los tres operadores siguientes se definen ahora como operadores básicos del proceso:

1) Selección de caso

Predicado $\varphi: E \rightarrow \{0,1\}$

Selección de caso: $CS_{\phi}: E \rightarrow E'$

$$E' := \left\{ \bigcup_i O_i \mid O_i \in E \wedge \phi(O_i) = 1 \right\}$$

5 Entre los ejemplos de operadores de selección de casos se incluye el Operador Coincidir para filtrar las instancias del proceso que coinciden con un patrón de proceso determinado.

2) Selección de corte

Predicado inicial $\rho_s: R \rightarrow \{0,1\}$

Predicado final $\rho_e: R \rightarrow \{0,1\}$

10 Caso Corte $CC_{\rho_s, \rho_e}: O_i \rightarrow O'_i$

$$O'_i = (\{e \mid e \in E_i \wedge \exists e_s \in E_i: \rho_s(e_s) = 1 \wedge \exists e_e \in E_i: \rho_e(e_e) = 1 \wedge e_s \leq e \leq e_e\}, Q_i)$$

Selección de corte $CS_{\rho_s, \rho_e}: E \rightarrow E'$

$$E' = \{ CC_{\rho_s, \rho_e}(O_i) \mid O_i \in E \}$$

15 Un ejemplo de ello, consiste en cortar una sección de un protocolo del proceso entre un comienzo definido y una actividad final definida para obtener un nuevo protocolo de subproceso.

20 Las ampliaciones en este caso son que el predicado se cumple por primera vez y se cumple por última vez. También se pueden extraer varios subflujos a través del CROP y solo se deben incluir la primera y la última actividad en el conjunto de resultados.

La relación de orden se mantiene con este predicado. Esto significa que se crea un nuevo protocolo del proceso utilizando este operador.

3) Proyección de eventos

25 Proyección $\delta: R \rightarrow R \circ R$

$$\delta(r) = r \circ r_n \mid r \in E_i \wedge r_n \in E_i \wedge r < r_n \wedge \exists r_2: r < r_2 < r_n \text{ sobre la relación de orden } Q_i$$

$$\gamma(O_i) = (\{\delta(e) \mid e \in E_i\}, \{\delta(q_s, q_e) \mid (q_s, q_e) \in Q_i\})$$

Proyección de eventos $\Omega: E \rightarrow E'$

$$\Omega(E) = \{ \gamma(O) \mid O \in E \}$$

35 Esta proyección crea un nuevo protocolo del proceso a partir de un protocolo del proceso en el que se concatenan entradas adyacentes. Un ejemplo de aplicación de esto es el uso para el cálculo de los tiempos de ejecución.

40 Sobre la base de la ampliación del álgebra relacional al álgebra de procesos de acuerdo con la invención, se puede cartografiar un gran número de operadores del proceso en APE utilizando una combinación de operadores básicos del proceso. Además, todos los operadores definidos en el álgebra relacional siguen estando disponibles. De esa forma, se pueden combinar libremente los operadores del proceso y los operadores de bases de datos convencionales.

45 El motor APE está adaptado para recibir declaraciones de APE de un cliente, procesar las declaraciones de APE recibidas y proporcionar los resultados del procesamiento. Si fuera necesario, los resultados también se pueden procesar posteriormente antes de ser devueltos al cliente.

50 De manera alternativa o adicional, los resultados se pueden poner a disposición del motor APE, por ejemplo, para realizar un análisis más detallado con base en los resultados (por ejemplo, para reducir aún más los resultados). También en este caso, los resultados se pueden procesar posteriormente antes de ponerlos a disposición del motor APE.

A modo de ejemplo, veamos un protocolo de un proceso de compra con la siguiente estructura:

ID de Orden	Evento	Marca de tiempo	Proveedor
1	Crear Orden de Compra	06.01.2012 11:01	ABC Machine Inc
1	Aprobar Orden de Compra	07.03.2012 14:07	ABC Machine Inc
1	Ingresar Recibo de Mercancías	01.04.2012 16:05	ABC Machine Inc
1	Ingresar Recibo de Orden de Compra	01.04.2012 17:02	ABC Machine Inc

1	Pago	03.05.2012 11:17	ABC Machine Inc
2	Crear Orden de Compra	14.02.2012 04:48	Fabricar CDF
2	Reprocesar Orden de Compra	20.03.2012 07:26	Fabricar CDF
2	Rechazar Orden de Compra	24.04.2012 01:26	Fabricar CDF
3	Crear Orden de Compra	29.04.2012 15:21	HIJ Insurance Corp
3	Aprobar Orden de Compra	30.04.2012 17:16	HIJ Insurance Corp
3	Ingresar Recibo de Orden de Compra	30.04.2012 23:31	HIJ Insurance Corp
3	Pago	01.05.2012 08:38	HIJ Insurance Corp
4	Crear Orden de Compra	06.02.2012 17:16	ABC Machine Inc
4	Aprobar Orden de Compra	28.03.2012 07:12	ABC Machine Inc
4	Ingresar Recibo de Mercancías	31.03.2012 08:52	ABC Machine Inc
4	Ingresar Recibo de Orden de Compra	05.04.2012 19:55	ABC Machine Inc
4	Pago	10.04.2012 14:52	ABC Machine Inc
4	Ingresar Recibo de Orden de Compra	20.04.2012 23:02	ABC Machine Inc

Las etapas o actividades de los procesos se almacenan en el protocolo del proceso al que tiene acceso el motor APE. Para cada etapa del proceso, el protocolo del proceso también puede contener información sobre el proveedor correspondiente. Sin embargo, resulta ventajoso que el protocolo del proceso solo contenga referencias a la información del proveedor, en donde la información del proveedor puede almacenarse en una fuente de datos externa (vea la Figura 2).

A continuación se presenta un ejemplo de una declaración APE (Declaración en Anotación APE), que se puede utilizar para calcular un plazo medio de ejecución de los procesos entre las actividades "Crear Orden de Compra" y "Pago" para todos los proveedores, por lo que entre estas dos actividades se puede dar cualquier otra cantidad de actividades. Dado que las actividades pueden ocurrir potencialmente varias veces en una instancia del proceso, consideramos la primera ocurrencia de la actividad "Crear Orden de Compra" y consideramos todos los tiempos de ejecución para cualquier ocurrencia de la actividad de "Pago".

```
TABLE ("VENDOR",
      MEDIAN (CALC_THROUGHPUT (
                REMAP_TIMESTAMPS ("TIMESTAMP", DAYS),
                FIRST_OCCURRENCE ['Create Purchase Order'] TO ANY_OCCURRENCE ['Payment']
            )
        )
    )
```

Los "tiempos de ejecución" de los operadores del proceso se pueden aplicar en el motor APE mediante una combinación de varios operadores básicos del proceso:

- (Operador del proceso) El protocolo del proceso se reduce mediante una selección CROP a la primera actividad "Crear orden de Compra" y a la última actividad "Pago". El predicado inicial es "Actividad" = 'Crear Orden de Compra', el predicado final es este caso "Actividad" = 'Pago'.
- (Operador DB) Será el atributo "TIMESTAMP" de una marca de tiempo por día.
- (Operador del proceso) Se ejecuta una proyección de eventos. Como resultado, se forma una nueva relación en la que se puede acceder al evento precedente y al sucesor como una tupla de la relación de resultados.
- (Operador DB) La marca de tiempo por día del evento sucesor se resta de la marca de tiempo por día del evento precedente.
- (Operador DB) Se ejecuta una agrupación de los plazos calculados de acuerdo con el atributo "VENDOR" con el agregador "MEDIAN".

Por tanto, de acuerdo con la invención, es posible combinar operadores aplicados a los protocolos del proceso con operadores aplicados a las tablas convencionales.

En este ejemplo, el resultado de la consulta precedente se da en forma de tabla, por ejemplo para mostrarla al cliente como una tabla. El resultado también se puede dar en un formato diferente, por ejemplo, en forma de gráfico, de diagrama, de flujo de proceso, etc.

Por consiguiente, en este caso, el resultado sería el siguiente:

Proveedor	Mediana
ABC Machine Inc	83
HIJ Insurance Corp	2

5

Veamos algunos ejemplos más de consultas APE.

10

La Figura 5 muestra un ejemplo de un proceso visualizado. Por ejemplo, el protocolo del proceso almacena 50 millones de actividades, cada una de las cuales se asigna a un proceso.

15

Con la ayuda de una primera consulta APE, se determinan ahora los tiempos medios de ejecución de todos los procesos que empiezan con la actividad "Crear Orden de Compra" y terminan con la actividad "Pago" y que tienen una cobertura del 20 %. Esto significa que se consultan las variantes del proceso en el protocolo del proceso que, en conjunto, representan el 20 % de todos los procesos en el protocolo del proceso. Para realizar este cálculo se puede utilizar la siguiente consulta APE:

20

```

FILTER
    MATCH_PROCESS (
        STARTING ["Create Purchase Order"]AS "Start",
        ENDING["Payment"]AS "End"
        CONNECTED BY EVENTUALLY ["Start", "End"]
    ) = 1;
VARIANT_TABLE =
TABLE (
VARIANT () AS V,
CUME_DIST () AS R
);
30 FILTER VARIANT_TABLE.R
< 0.2;
GRAPH ();

```

25

30

35

En este ejemplo, el protocolo del proceso se restringe primero a los casos que comienzan con la actividad "Crear Orden de Compra" y terminan con la actividad "Pago".

Para ello se utiliza el operador del proceso "MATCH_PROCESS".

40

En la segunda etapa, se determina una lista temporal de las variantes del proceso y la distribución acumulativa para cada variante del proceso. Para ello se utiliza operador del proceso VARIANT() y el Operador DB CUME_DIST().

45

De acuerdo con la invención, es posible utilizar los resultados de los análisis para un filtrado posterior. Para ello, el motor APE recibe una referencia entre las relaciones fuente y los resultados correspondientes. En el ejemplo, se crea el enlace entre la tabla VARIANT_TABLE agregada a nivel de variante y el protocolo del proceso original. Esto permite restringir el protocolo del proceso a aquellos casos cuya variante se encuentra entre el 20 % más frecuente en la siguiente etapa (distribución acumulativa < 0,2).

50

Para mostrar el gráfico del proceso, el operador del proceso GRAPH() calcula y entrega un gráfico basado en el protocolo del proceso filtrado.

55

El gráfico del proceso para el resultado de esta primera consulta se muestra en la Figura 5 a). El gráfico del proceso también calcula y muestra la frecuencia de ejecución de los bordes. Esta información no se puede obtener utilizando las posibilidades conocidas en el estado de la técnica.

60

El usuario ahora cambia el grado de cobertura a 40 %. Con la ayuda de una segunda consulta APE, se determinan ahora los tiempos medios de ejecución de todos los procesos que empiezan con la actividad "Crear Orden de Compra" y terminan con la actividad "Pago" y que tienen una cobertura de 40 %. Gracias a la estructura de datos de acuerdo con la invención (protocolo del proceso disponible como flujo de proceso) y al lenguaje de consulta APE de acuerdo con la invención, el resultado se visualiza en tiempo real en el dispositivo de visualización del usuario. El gráfico del proceso para el resultado de esta segunda consulta se muestra en la Figura 5 (b). El 40 % de todos los procesos en el protocolo del proceso corresponden a esta variante del proceso. La variante de proceso más común se indica aquí con flechas en negrita, donde el grosor de las flechas es proporcional al número de procesos concretos.

65

Si se aumenta la cobertura a 100 %, se muestran todas las diferentes variantes de los procesos almacenados en el protocolo del proceso.

De acuerdo con la invención, el usuario tiene la posibilidad, por ejemplo, de aplicar ciertos filtros a las variantes del proceso visualizado. Por ejemplo, puede seleccionar una actividad (por ejemplo, "Envío") y seleccionar todas las variantes de proceso del protocolo del proceso que se ejecutan sobre la actividad seleccionada. La consulta APE precedente tendría que ser ampliada por el siguiente filtro:

5 FILTER MATCH_PROCESS (NODE ["Shipping"]) = 1

A continuación se describen con más detalle la arquitectura y la estructura de los datos de acuerdo con la invención.

10 Tomando como referencia la Figura 1, los clientes pueden acceder al servidor, por ejemplo, a través de Internet o de una intranet. Por ejemplo, un cliente puede ser un ordenador con un cliente de Internet. Alternativamente, también se puede prever como cliente, una aplicación, por ejemplo una hoja de cálculo, que accede al servidor a través de una interfaz. Un cliente también se puede diseñar como un dispositivo móvil.

15 El servidor puede tener un servidor WEB (por ejemplo, Apache HTTP Server o Microsoft Internet Information Server), destinado a manejar la transferencia de datos entre un cliente de Internet y el servidor.

El servidor también incluye el motor APE descrito anteriormente. El motor APE está adaptado para recibir consultas APE y convertirlas o transformarlas en una forma ejecutable. La forma ejecutable de una consulta APE puede ser un programa o secuencia de comandos ejecutable. Al servidor o al motor APE se pueden subordinar memorias caché, para poder almacenar eficientemente resultados intermedios o parciales cuando se ejecuta el programa o script ejecutable.

20 Los datos necesarios para el protocolo del proceso se pueden cargar como parte de un proceso ETL desde uno o más sistemas fuente (CRM 1 a CRM n) y almacenarse como un protocolo del proceso de acuerdo con una estructura de datos de acuerdo con la invención. La estructura de los datos se describe con más detalle a continuación.

Preferentemente, los datos o el protocolo del proceso se almacenan en la memoria de trabajo (por ejemplo, como una base de datos en memoria). Esto mejora considerablemente el desempeño al evaluar los procesos. Los sistemas fuente pueden incluir, por ejemplo, sistemas ERP estandarizados (por ejemplo, de SAP AG, Walldorf, Alemania) o sistemas patentados que almacenan instancias del proceso de uno o más procesos comerciales.

30 Resulta ventajoso que el protocolo del proceso se divida en varias partes de aproximadamente del mismo tamaño, en donde los procesos siempre se almacenan completos en una parte. Esto permite dividir el protocolo del proceso, por lo que las particiones se pueden procesar en paralelo.

35 En la Figura 2 (a) se muestra un ejemplo de partición. Aquí el protocolo del proceso se divide en tres particiones de aproximadamente el mismo tamaño, cada una de las cuales se mantiene en la memoria.

40 Al cargar desde los sistemas fuente, se cargan todos los datos necesarios para el protocolo del proceso. Alternativamente, los datos también se pueden cargar de forma incremental. Esto significa que solo se cargan los datos que han cambiado o se han añadido en los sistemas fuente desde la última carga.

45 Si no es posible el acceso directo a uno o más sistemas fuente, se puede prever una interfaz IF para acceder a los sistemas fuente correspondientes.

En la Figura 2 muestra una estructura de datos de un protocolo del proceso de acuerdo con la invención.

50 En su forma más simple, la estructura de datos del protocolo del proceso tiene tres atributos (Felder), a saber, "ID de Caso", "Actividad" y "Orden".

El atributo "ID de Caso" es un identificador único de una instancia del proceso (por ejemplo, un proceso comercial concreto). El atributo "Actividad" describe una etapa del proceso de una instancia del proceso. El atributo "Orden" especifica la secuencia de etapas del proceso dentro de una instancia del proceso.

55 El atributo "ID de Caso" se utiliza para asignar todos las etapas del proceso a una instancia concreta del mismo. En una modalidad de la invención, el atributo "ID de Caso" puede ser un identificador único compuesto (clave compuesta).

60 El atributo "Actividad" puede almacenar un identificador de la etapa del proceso. El identificador puede ser, por ejemplo, una descripción de la etapa del proceso.

Los valores de los atributos se pueden almacenar en forma comprimida, por ejemplo, mediante un procedimiento de codificación de diccionario.

65 En general, el atributo "Orden" especifica la secuencia de etapas del proceso dentro de una instancia del proceso. En una modalidad de la invención, se puede almacenar una hora (fecha/hora) en el atributo "Orden", que especifica el momento

en que se ejecutó la etapa del proceso correspondiente. El uso de horarios (por ejemplo, las marcas de tiempo) tiene la ventaja de que se pueden calcular los tiempos de ejecución de los procesos.

5 De acuerdo con la invención, se pretende ordenar el protocolo del proceso o los protocolos de datos almacenados en él primero de acuerdo con el atributo "ID de Caso" y luego de acuerdo con el atributo "Orden".

10 Los datos fuente (Datos de Proceso Externo), que también incluyen datos de proceso y se pueden almacenar en una base de datos o de otra manera en un sistema externo (Sistemas Fuente), se cargan en la memoria de trabajo y luego se clasifican físicamente en la memoria de trabajo como se ha explicado anteriormente. Esto significa que las etapas del proceso pertenecientes a un proceso se almacenan en la memoria de trabajo en direcciones adyacentes.

En lo adelante, "protocolo del proceso" se refiere siempre al protocolo del proceso clasificado almacenado en la memoria de trabajo. Este protocolo del proceso también se denomina en lo adelante flujo de proceso.

15 La clasificación de las etapas del proceso dentro de una instancia del proceso tiene la ventaja de que las instancias del proceso están disponibles en forma agrupada. Esto permite que las etapas del proceso en un flujo se procesen secuencialmente y en el orden correcto. También resulta ventajoso poder calcular una relación precedente/sucesor entre las etapas del proceso dentro de una instancia del proceso.

20 A continuación se muestra un ejemplo de un protocolo del proceso.

ID de Caso	Actividad	Orden
1	Crear Orden de Compra	08:00
1	Envío	10:00
1	Pago	11:00
2	Crear Orden de Compra	08:00
2	Aprobar Orden de Compra	08:15
2	Envío	08:30
2	Pago	09:00

35 El protocolo del proceso se almacena en la memoria de trabajo de acuerdo con la invención, es decir, el contenido del protocolo del proceso se almacena en columnas.

40 La estructura de datos de acuerdo con la invención del protocolo del proceso y la clasificación de acuerdo con la invención de los protocolos de datos tienen la ventaja de que la reconstrucción de los procesos y las operaciones que se les aplican se puede llevar a cabo en tiempo lineal con el motor APE de acuerdo con la invención descrito anteriormente y el lenguaje de consulta APE descrito más detalladamente a continuación tomando como referencia la Figura 4.

45 Además o alternativamente, la estructura de datos del protocolo del proceso puede tener una o más referencias (claves externas) a datos externos. Estas referencias se pueden almacenar en uno o más atributos "FK". Esto permite enriquecer el resultado de un análisis del proceso con datos adicionales. Alternativa o adicionalmente, los procesos también se pueden filtrar utilizando los datos externos.

50 Los datos externos son los que no se almacenan en el protocolo del proceso. Estos datos se pueden guardar en el mismo sistema que el protocolo del proceso. En una modalidad de la invención, los datos externos también se pueden almacenar en la memoria de trabajo. Un ejemplo de datos externos son los datos maestros de los clientes que participan en los procesos.

55 La Figura 4 muestra un diagrama de bloques con un motor APE de acuerdo con la invención. Este diagrama de bloques se utiliza para describir más detalladamente una posible configuración del lenguaje de consulta APE de acuerdo con la invención.

60 En una primera etapa, la declaración APE recibida se pasa a un analizador del compilador APE. El analizador analiza la declaración APE y genera un árbol de sintaxis abstracta (AST) a partir de ella, que se muestra a continuación en forma simplificada para la declaración APE precedente.

```

65 TABLE <CalculationType>
    VENDOR <Column>
    MEDIAN <Aggregate>
        CALC_THROUGHPUT (ANY ...) <ThroughputCalculation>
    
```

El árbol de sintaxis abstracta representa una secuencia lógica de las operaciones u operadores individuales de la declaración APE.

5 El árbol de sintaxis abstracta se ejecuta o procesa en un entorno de ejecución en memoria. Puede ser ventajoso convertir el árbol de sintaxis abstracta antes de la ejecución en un formato que sea útil para el entorno de ejecución en memoria. También puede ser ventajoso optimizar el árbol de sintaxis abstracta antes de la ejecución, por ejemplo para reducir el tiempo de ejecución.

10 Cuando el árbol de sintaxis abstracta se ejecuta en el entorno de ejecución en memoria, primero se genera un plan de ejecución utilizando el árbol de sintaxis abstracta, que luego se ejecuta.

A continuación se describe un ejemplo de cómo se crea el plan de ejecución.

15 El plan de ejecución especifica las etapas y la secuencia en la que el motor APE ejecuta las operaciones u operadores.

Al generar el plan de ejecución, se extraen primero todos los operadores del árbol de sintaxis y se ponen a disposición como referencias en el plan de ejecución.

20 El plan de ejecución del árbol de sintaxis precedente comprende los siguientes elementos:

1. Operadores

OP 1:

Tipo de Operador: COLUMNA

Ref. Operador ID: 1

25 Nombre Columna: PROVEEDOR

OP 2:

Tipo de Operador: Cálculo de Rendimiento

Ref. Operador ID: 2

Inicio: ANY ...

30 Fin: ANY ...

Unidades de Tiempo: Días

2. Agregar (Plan de ejecución)

GROUPERS: OP_REF 1

AGGREGATORS:

35 Función de Agregación: MEDIAN

Operador de Función de Agregación: OP_REF 2

"ID Ref. Operador" es una identificación única del operador correspondiente con la que se puede hacer referencia al operador en el plan de ejecución (mediante "OP_REF").

40 En el ejemplo anterior, el operador "OP 1" del plan de ejecución especifica que los resultados deben agruparse por proveedor (Nombre Columna: VENDOR) (GROUPERS: OP_REF 1). El operador "OP 2" especifica en el plan de ejecución que se debe calcular un tiempo de ejecución (Cálculo de Rendimiento) entre dos etapas del proceso (Inicio/Fin).

45 Es importante que el operador "OP 2" se considere como una columna normal de una tabla, aunque se calcule utilizando el flujo de proceso.

La ejecución del plan de ejecución se describe en el siguiente ejemplo:

50 Al ejecutar el plan de ejecución, se ejecutan primero todos los operadores (en el ejemplo anterior, los operadores "OP 1" y "OP 2").

Para el operador "OP 1", simplemente se devuelve una referencia a la columna "VENDOR". La columna "VENDOR" contiene información sobre los correspondientes proveedores.

55 Para el operador "OP 2", que aquí se denomina "Operador de Rendimiento", se genera un código de programa (ejecutable) y se ejecuta. Alternativamente, se ejecuta un código de programa ya existente. En este ejemplo, el código de programa se adapta para calcular los tiempos de ejecución entre las actividades "Crear Orden de Compra" y "Pago" para los procesos almacenados en el protocolo del proceso.

60 Una vez ejecutados los dos operadores "OP 1" y "OP 2", se dispone de dos referencias a las columnas, a saber, una referencia a la columna "VENDOR" y una referencia a los tiempos de ejecución calculados.

Después, se pueden aplicar filtros a las columnas en una etapa opcional, por ejemplo, para asegurar que solo se tengan en cuenta en las etapas de cálculo sucesor los protocolos de datos que cumplan un determinado criterio de filtro.

65

Una vez ejecutados los operadores, las dos referencias a las columnas generadas se transfieren a un llamado autómata de agregación en el siguiente etapa. El autómata de agregación se adapta para agrupar los tiempos de ejecución calculados por proveedor y para determinar un plazo medio de entrega para cada proveedor.

5 Después que el motor APE también ejecuta esta última etapa, se devuelve al cliente el resultado de la declaración APE o la consulta APE.

10 Alternativa o adicionalmente, el resultado del motor APE puede estar disponible como entrada para otras consultas APE. Esto permite realizar consultas anidadas. En este caso, resulta ventajoso que el resultado del motor APE se proporcione como un flujo de proceso.

15 Lo precedente es un ejemplo de una consulta APE en la que el operador del proceso es el tiempo de ejecución. Lo precedente es un ejemplo de una consulta APE en la que el operador del proceso es el tiempo de ejecución. Ejemplos de estos operadores del proceso se muestran tomando como referencia la Figura 1.

Como se puede apreciar en el ejemplo precedente de una consulta APE, en una consulta APE, los operadores del proceso también pueden combinarse con operadores aplicados a tablas convencionales y/o con operadores DB convencionales (por ejemplo, agregaciones, uniones, etc.).

20 El cliente recibe el resultado de la consulta APE y puede visualizarlo en un dispositivo de visualización. Por ejemplo, si el resultado se devuelve en forma de gráfico, se puede visualizar como un gráfico del proceso.

25 De acuerdo con la invención, el cálculo de los tiempos de ejecución se puede realizar sobre la base de uno o más calendarios, como se explica más detalladamente a continuación.

30 El cálculo de los tiempos de ejecución, especialmente los tiempos de ejecución netos, es de gran importancia para un análisis exploratorio de los protocolos del proceso. En el estado actual de la técnica, los tiempos de ejecución se calculan siempre como tiempos de ejecución brutos, lo que tiene la desventaja de que el tiempo de ejecución calculado no se corresponde con el tiempo de ejecución relevante comercialmente (por ejemplo, los horarios de servicio, los horarios de entrega en días laborables, etc.).

35 A continuación se presenta un ejemplo de método para el cálculo ad hoc de cualquier tiempo de ejecución neto: Cuando se calculan los tiempos de ejecución basados en uno o más calendarios, es posible, de acuerdo con la invención, que no se tengan en cuenta los días festivos, los fines de semana o determinados momentos del día.

De acuerdo con la invención, el método descrito a continuación se utiliza para determinar los tiempos de ejecución netos entre las etapas del proceso.

40 El método para calcular los tiempos de ejecución netos de acuerdo con la invención comprende esencialmente dos etapas, en donde en la primera etapa se transforman (normalizan) los datos de tiempo almacenados en el protocolo del proceso para las etapas del proceso en relación con una base, y en la segunda etapa se determina una diferencia entre los datos de tiempo transformados asignados a las etapas del proceso. Los datos de tiempo pueden ser marcas de tiempo.

45 Para transformar o normalizar las marcas de tiempo, todas las marcas de tiempo, preferentemente todos las marcas de tiempo diferentes, se clasifican primero en orden ascendente en el protocolo del proceso. Este vector (denominado en lo adelante vector de marca de tiempo) de las marcas de tiempo clasificadas en orden ascendente constituye el punto de partida para determinar un desfase para cada marca de tiempo (diferente) en el protocolo del proceso.

50 Cada marca de tiempo puede consistir en una fecha (por ejemplo, 31/05/2014) e información sobre la hora (por ejemplo, 13:37:00) con una precisión dada (por ejemplo, en segundos).

55 Las marcas de tiempo se transforman o normalizan ahora de manera que la diferencia entre cada dos marcas de tiempo transformadas sucesivamente es la diferencia neta de su diferencia de tiempo en la unidad de destino (por ejemplo, días, horas, ...).

Para transformar las marcas de tiempo se puede especificar lo siguiente:

- Para cada día (fecha), hay una definición de si se tiene en cuenta o no. Esta definición se puede hacer, por ejemplo, especificando los días de la semana (por ejemplo, lunes) o especificando explícitamente las fechas utilizando un calendario (por ejemplo, 24/05/2016). Tal definición también se puede concebir por semanas, meses, trimestres, etc.
- Para cada día hay una definición de las horas del día que se deben considerar o no. Esto se puede hacer, por ejemplo, especificando las horas de un día de la semana (por ejemplo, lunes: 08:00 a 17:00) o especificando explícitamente las horas en una fecha determinada utilizando un calendario (por ejemplo, 24/05/2016: 08:00 a 17:00). Tal definición también se puede concebir por semanas, meses, trimestres, etc. Por ejemplo, se puede definir semanalmente qué días de la semana se deben considerar, o mensualmente qué días se deben considerar.
- La duración neta máxima de un día es la suma de los intervalos del día activos de ese día.

•
También se pueden definir varios calendarios que se pueden tener en cuenta simultáneamente. Por ejemplo, se puede utilizar un primer calendario para especificar las semanas de un año que se van a considerar, mientras que un segundo calendario se puede utilizar para especificar los días de este año que no se van a considerar. Por ejemplo, se pueden definir excepciones diariamente para las semanas especificadas en el primer calendario.

Para facilitar el procesamiento, la fecha se puede utilizar como un número continuo para cada marca de tiempo (por ejemplo, la fecha juliana) y la hora se puede utilizar como dato de tiempo con suficiente precisión (por ejemplo, milisegundos desde las 00:00 horas).

Si la diferencia se calcula en días, se puede ignorar el componente de la hora de la marca de tiempo y se puede realizar el cálculo solo sobre el componente de la fecha.

Para transformar la marca de tiempo, también se requiere la duración neta del dato de tiempo en una unidad predeterminada (por ejemplo, horas, minutos o segundos). Por ejemplo, la duración neta corresponde a las horas netas (o minutos netos o segundos netos) de la marca de tiempo desde las 00:00 horas.

Aquí va un ejemplo: Si se define un intervalo de tiempo a tener en cuenta para un día de 08:00 a 17:00, la duración neta en base a minutos desde las 03:00 horas es de 0 minutos, al igual que la duración neta desde las 07:00 horas. La duración neta desde las 09:00 es de 60 minutos. La duración neta desde las 17:00 es de 540 minutos. La duración neta desde las 23:00 horas es de 540 minutos y por lo tanto corresponde a la máxima duración neta de este día, porque no se tiene en cuenta el intervalo de 17:00 a 23:00 horas.

El tiempo neto total diario es el tiempo neto de un día entero.

La transformación de las marcas de tiempo se realiza ahora de acuerdo con el siguiente procedimiento:

En una primera etapa, se realiza una inicialización, que incluye

- el vector resultante se inicializa como un vector cero con la misma dimensión que el vector de la marca de tiempo;
- el desfase del día actual se inicializa con el desfase del día de la primera marca de tiempo;
- el desfase actual se inicializa con la duración neta del primer marca de tiempo;
- el valor actual de compensación se inicializa con 0; y
- al vector resultante se le asigna el valor de desfase actual.

•
En una segunda etapa, se realiza el siguiente procedimiento de forma iterativa para todas las marcas de tiempo en el vector de marcas de tiempo:

- siempre y cuando el desfase del día de la próxima marca de tiempo sea mayor que el desfase del día actual:
 - aumentar el valor del desfase en la duración neta del día actual (si no se tiene en cuenta el día, es 0) menos el desfase de la hora actual;
 - aumentar el desfase del día actual en 1; y
 - ajustar el desfase de la hora actual a 0;
- el desfase del día de la próxima marca de tiempo es igual al desfase del día actual:
 - aumentar el valor de desfase actual en la diferencia entre el desfase de la hora actual y la duración neta de la próxima marca de tiempo;
 - asignar el valor de desfase actual al vector de resultado en la posición de la próxima marca de tiempo; y
 - comenzar con la próxima marca de tiempo.

○
Como resultado, usando las marcas de tiempo transformadas de esta manera, se pueden calcular los tiempos netos entre dos marcas de tiempo cualquiera del conjunto de todas las marcas de tiempo.

En una modalidad ventajosa de la invención, los datos de tiempo de las etapas del proceso del protocolo del proceso se almacenan en una codificación de diccionario. El procedimiento para determinar la duración neta se puede realizar directamente sobre los datos de tiempo codificados en el diccionario y, por lo tanto, sobre los datos comprimidos.

En otra modalidad ventajosa de la invención, los valores en el diccionario están ordenados en orden ascendente. Esto elimina la necesidad de una clasificación previa para determinar la duración neta.

En el caso más sencillo, se puede calcular el tiempo de ejecución, es decir, el tiempo de ejecución neto de una instancia del proceso completo entre la primera y la última etapa del proceso de la instancia del proceso. Sin embargo, también es posible calcular el tiempo de ejecución neto de un subproceso de la instancia del proceso, es decir, el tiempo de ejecución neto entre las etapas del proceso de la instancia del proceso que son diferentes de la primera etapa del proceso o diferentes de la última etapa del proceso.

El usuario puede seleccionar las etapas del proceso entre las cuales se calculará el tiempo de ejecución neto en una interfaz de usuario.

Alternativa o adicionalmente, la selección de las etapas del proceso entre las que se debe calcular el tiempo de ejecución neto también se puede realizar utilizando reglas. Ejemplos de esto son:

- Primera ocurrencia de una etapa del proceso específica dentro de una instancia del proceso;
- Última ocurrencia de una etapa del proceso específica dentro de una instancia del proceso;
- Cualquier ocurrencia de una etapa del proceso particular dentro de una instancia del proceso;
- Primera ocurrencia de cualquier etapa del proceso dentro de una instancia del proceso;
- Última ocurrencia de cualquier etapa del proceso dentro de una instancia del proceso;
- Etapa del proceso con un identificador predeterminado.

Esta lista de ejemplos de selección de etapas del proceso utilizando reglas no es exhaustiva. También se pueden utilizar otras reglas para seleccionar las etapas del proceso.

Esos tiempos de ejecución netos se pueden calcular para varias instancias del proceso. En particular, esos tiempos de ejecución netos se pueden calcular para todas las instancias del proceso que figuran en el protocolo del proceso. A partir de los tiempos de ejecución netos calculados, se puede calcular entonces, por ejemplo, un tiempo de ejecución neto medio para varias o todas las instancias del proceso de un protocolo del proceso. En otro ejemplo, también se pueden seleccionar aquellas instancias del proceso del protocolo del proceso cuyo tiempo de ejecución neto cumple un criterio predefinido (por ejemplo, cuyo tiempo de ejecución neto es inferior a 5 días).

A continuación se explica cómo se pueden reconocer los subprocesos de instancias del proceso que funcionan en paralelo.

Un proceso puede constar de varios subprocesos, en donde los subprocesos se pueden ejecutar en paralelo. Los subprocesos, a su vez, también pueden estar compuestos por subprocesos que también se pueden haber ejecutado en paralelo. En la mayoría de los casos, no es posible detectar subprocesos que se ejecuten en paralelo utilizando únicamente el atributo "Orden". Sin embargo, la detección de subprocesos que ocurren en paralelo es necesaria para la correcta determinación de los gráficos o jerarquías de los procesos por medio de operadores del proceso, lo que no se puede garantizar con los métodos conocidos en el estado de la técnica.

De acuerdo con la invención, se proporciona para este fin una forma ampliada de la estructura de datos del protocolo del proceso, que también se muestra en la Figura 2. Esta forma ampliada de la estructura de datos tiene solo dos atributos adicionales, a saber, "ID de Padre" y "ID de Subproceso".

El atributo "ID de Subproceso" almacena un identificador único para un subproceso de un proceso o para un subproceso de un subproceso. El atributo "ID de Padre" especifica a qué proceso matriz pertenece un subproceso. De acuerdo con la invención, estos dos atributos se utilizan para describir una jerarquía de procesos, en donde los subprocesos dentro de la jerarquía se pueden haber ejecutado en paralelo. Este paralelismo siempre se reconoce, independientemente de la profundidad de la jerarquía.

De acuerdo con la invención, también es posible determinar los valores de los atributos "ID de Padre" y "ID de Subproceso" sobre la base de otras características contenidas en los datos. La información contenida en los atributos "ID de Padre" y "ID de Subproceso" también se puede almacenar en una forma o estructura diferente.

De acuerdo con la invención, la forma extendida de la estructura de datos permite calcular una división de un proceso en subprocesos paralelos y una fusión de subprocesos paralelos en tiempo lineal. De esta manera, los subprocesos paralelos también se pueden analizar eficientemente y, en particular, correctamente. Las pruebas han demostrado que todos los puntos de un proceso, en los que el proceso se divide en subprocesos paralelos o en los que se fusionan subprocesos paralelos, se pueden calcular en una fracción de segundo para protocolos del proceso con varios millones de entradas de datos. El cálculo de la jerarquía de los procesos o la determinación de los procesos que se ejecutan en paralelo se describe más detalladamente a continuación tomando como referencia la figura 3.

Si la profundidad de la jerarquía es de 1 como máximo (es decir, si un proceso tiene subprocesos, pero estos subprocesos en sí mismos no tienen subprocesos), se puede omitir el atributo "ID de Padre".

La Figura 3 muestra un ejemplo de un protocolo del proceso (Figura 3(a)), una jerarquía de estados establecida a partir de él (Figura 3(b)) y un gráfico correspondiente (Figura 3(c)).

En la Figura 3 se describe más detalladamente la determinación de los subprocesos paralelos y la determinación de la jerarquía de procesos basada en el protocolo del proceso.

En la Figura 3 (a) se muestra un protocolo del proceso en el que se almacenan seis actividades (etapas del proceso) A a F para el proceso con el ID de Caso = 1. Este proceso comprende dos subprocesos que se ejecutan en paralelo. Los dos subprocesos no incluyen ningún otro subproceso, por lo que el atributo "ID de Padre" no es absolutamente necesario. Las actividades pertenecientes a los correspondientes subprocesos se identifican con los valores del atributo "IDSubProc".

5 En una primera etapa, se determina una jerarquía de estado a partir del protocolo del procesos. La jerarquía de estados determinada se muestra en la Figura 3 (b). La estructura de la jerarquía de estados tiene lugar ventajosamente en la memoria de trabajo, en donde el protocolo del proceso se procesa secuencialmente en una etapa de iteración. Por lo tanto, la jerarquía de estados se puede determinar con una complejidad temporal de $O(n)$.

La Figura 3 (b) explica cómo se genera la jerarquía de estados.

10 En una primera etapa (Etapa 1) se procesa el primer protocolo de datos (Orden = 1) del protocolo del proceso. Este primer protocolo de datos corresponde a la primera etapa del proceso. El valor del atributo "IDSubProc" asigna este primer protocolo de datos a un subproceso. En este caso, el subproceso es el proceso principal en sí mismo. Por lo tanto, el proceso superior no está establecido.

15 En la primera etapa (Etapa 1), el subproceso "Principal" se añade a la jerarquía de estados. Además, se asigna un indicador al primer protocolo de datos al subproceso "Principal", que también es el proceso principal. Esto significa que el protocolo de datos "1" está asignado al subproceso "Principal" en la jerarquía de estados.

20 En una segunda etapa (Etapa 2), se procesa el segundo protocolo de datos (Orden = 2) del protocolo del proceso. Este segundo protocolo de datos corresponde a la segunda etapa del proceso y también se asigna al subproceso "Principal" (= proceso principal). En la jerarquía de estados, el protocolo de datos "2" se asigna por lo tanto al subproceso "Principal".

25 Después de la segunda etapa (Etapa 2), se asignan dos valores al subproceso "Principal" en la jerarquía de estados. Debido a que el segundo valor añadido (2) es diferente al primer valor añadido (1), en la segunda etapa se añade un borde adicional "1→2" a la jerarquía de estados. Los bordes "x→y" añadidos en esta etapa y en las siguientes indican una relación precedente/sucesor, a saber, "y" es la sucesor de "x".

30 En una tercera etapa (Etapa 3) se procesa el tercer protocolo de datos (Orden = 3) del protocolo del proceso. Este tercer conjunto de datos corresponde a la tercera etapa del proceso. A diferencia de los dos primeros protocolos de datos, este tercer protocolo de datos se asigna al subproceso "SP1". El atributo "ID de Padre" del tercer protocolo de datos indica que está asignado al subproceso superior "Principal" (= proceso principal), es decir, el tercer protocolo de datos pertenece al proceso "SP1", que es un subproceso del proceso "Principal".

35 El subproceso "SP1" se añade a la jerarquía de estados en la tercer etapa (Etapa 3). Además, el protocolo de datos "3" se asigna al subproceso "SP1" en la jerarquía de estados. El protocolo de datos "2" asignado al subproceso "Principal" en la segunda etapa (Etapa 2) no ha cambiado y por lo tanto sigue asignado al subproceso "Principal".

40 Debido a que el valor "SP1" del atributo "IDSubProc" es diferente al valor "Principal" del atributo "ID de Padre" y a que no se asignó ningún valor al subproceso "SP1" en la segunda etapa (Etapa 2), el tercer protocolo de datos es la primera etapa del proceso del subproceso "SP1". Por lo tanto, el precedente de la tercera etapa del proceso es la etapa del proceso del proceso superior que se añadió por última vez a la jerarquía de estados. En este ejemplo, esta es la segunda etapa del proceso que se añadió en la segunda etapa (Etapa 2). En la tercera etapa, se añade un borde adicional "2→3" a la jerarquía del estado.

45 En una cuarta etapa (Etapa 4), se procesa el cuarto protocolo de datos (Orden = 4) del protocolo del proceso. Este cuarto protocolo de datos corresponde a la cuarta etapa del proceso. A diferencia del tercer conjunto de datos, este cuarto conjunto de datos se asigna al subproceso "SP2", asignado al subproceso superior "Principal" (= proceso principal), es decir, el cuarto conjunto de datos pertenece al proceso "SP2", que es un subproceso del proceso "Principal".

50 Debido a que tanto el subproceso "SP1" como el subproceso "SP2" están asignados al mismo proceso "Principal", los dos subprocesos "SP1" y "SP2" son procesos que se ejecutan en paralelo.

55 En la cuarta etapa, el subproceso "SP2" se añade a la jerarquía de estado. Además, el protocolo de datos "4" se asigna al subproceso "SP2" en la jerarquía de estados. El protocolo de datos "2" asignado al subproceso "Principal" en la tercera etapa (Etapa 3) no cambia y por lo tanto permanece asignado al subproceso "Principal". El protocolo de datos "3" asignado al subproceso "SP1" en la tercera etapa (Etapa 3) tampoco se modifica y, por lo tanto, sigue asignado al subproceso "SP1".

60 Debido a que el valor "SP2" del atributo "IDSubProc" es diferente al valor "Principal" del atributo "ID de Padre" y a que no se asignó ningún valor al subproceso "SP2" en la tercera etapa (Etapa 3), el cuarto conjunto de datos es la primera etapa del proceso del subproceso "SP2". Por lo tanto, el precedente de la cuarta etapa del proceso es la etapa del proceso del proceso superior que se añadió por última vez a la jerarquía de estados. En este ejemplo, esta es la segunda etapa del proceso que se añadió en la segunda etapa (Etapa 2). En la tercera etapa, se añade un borde adicional "2→4" a la jerarquía del estado.

65 En una quinta etapa (Etapa 5), se procesa el quinto protocolo de datos (Orden = 5) del protocolo del proceso. Este quinto protocolo de datos se asigna al subproceso "SP1", asignado al subproceso superior "Principal" (= proceso principal), es

decir, el quinto protocolo de datos pertenece al proceso "SP1", que es un subproceso del proceso "Principal". Además, el protocolo de datos "5" se asigna al subproceso "SP1" en la jerarquía de estados. El protocolo de datos "2" asignado al subproceso "Principal" en la cuarta etapa (Etapa 4) no cambia y por lo tanto permanece asignado al subproceso "Principal". El protocolo de datos "4" asignado al subproceso "SP2" en la cuarta etapa (Etapa 3) tampoco se modifica y, por lo tanto, sigue asignado al subproceso "SP2".

Dado que el subproceso "SP1" ya se ha añadido a la jerarquía de estados (en la tercera etapa (Etapa 3)), el quinto protocolo de datos es una etapa del proceso del subproceso "SP1" que no es la primera etapa del proceso del subproceso "SP1".

Por lo tanto, el precedente de la quinta etapa del proceso es la etapa del proceso del subproceso "SP1" que se añadió por última vez a la jerarquía de estados. En este ejemplo, esta es la tercera etapa del proceso que se añadió en la tercera etapa (Etapa 3). En la quinta etapa, por lo tanto, se añade un borde adicional "3→5" a la jerarquía del estado.

En una sexta etapa (Etapa 6), se procesa el sexto protocolo de datos (Orden = 6) del protocolo del proceso. Este sexto protocolo de datos se asigna al subproceso "Principal", que también es el proceso principal (ID de Padre = Principal).

El protocolo de datos "6" se asigna al subproceso "Principal". De la jerarquía de estados se desprende ahora que el protocolo de datos actual (= 6) del subproceso "Principal" es diferente al valor (= 2) asignado al subproceso "Principal" en la quinta etapa (Etapa 5). También se sabe que el subproceso "Principal" comprende varios subprocesos (en este ejemplo, SP1 y SP2). Por lo tanto, la sexta etapa del proceso es una etapa del proceso en la que se combinan los procesos subordinados (= subprocesos SP1 y SP2). La sexta etapa del proceso, por lo tanto, forma el llamado "punto de unión". Por esta razón, el valor actual (= 6) del subproceso "Principal" también se asigna a los subprocesos en la jerarquía de estado.

Por lo tanto, la sexta etapa del proceso tiene dos precedentes, a saber, las últimas etapas del proceso añadidas a los subprocesos SP1 y SP2. Por lo tanto, en la sexta etapa, se añaden adicionalmente los bordes, "5→6" y "4→6" de la jerarquía de estados.

Opcionalmente, se puede añadir otro borde "2→6" del estado anterior del subproceso "Principal".

De la jerarquía de estados creada de esta manera o de los bordes de jerarquía de estados, resulta ahora una jerarquía de procesos, que puede incluir subprocesos ejecutados en paralelo y que se puede visualizar, por ejemplo, como una estructura de árbol o como un gráfico.

La Figura 3 (c) muestra un gráfico generado a partir de la jerarquía de estados de acuerdo con la Figura 3 (b). Los subprocesos de ejecución paralela "SP1" y "SP2" se pueden reconocer aquí como partes del proceso principal "Principal". El borde "2→6" se muestra aquí como una flecha discontinua.

Este método tiene la ventaja de que la jerarquía de un proceso que comprende trayectos de proceso ejecutados en paralelo se puede determinar con una complejidad temporal de $O(n)$, independientemente de la profundidad de la jerarquía del proceso.

Debido a que los trayectos paralelos dentro de un proceso también se pueden determinar correctamente, siempre se obtiene el resultado correcto al realizar un análisis de procesos con trayectos de procesos paralelos. Por ejemplo, se puede determinar un tiempo medio de ejecución para los procesos que comienzan con la etapa del proceso "A" y en los que las etapas del proceso "C" y "D" se ejecutan en paralelo. Con los métodos conocidos en el estado de la técnica, no siempre se puede garantizar que las etapas del proceso que se ejecutan en paralelo se reconozcan realmente como tales. Por ejemplo, al determinar un tiempo medio de ejecución de los procesos que comienzan con la etapa del proceso "A" y en los que se ejecutan en paralelo las etapas del proceso "C" y "D", no se tienen en cuenta todos los procesos ejecutados, aun cuando en realidad cumplirían el criterio de filtro. Esto lleva inevitablemente a un tiempo medio de ejecución incorrecto.

La ventaja esencial de la invención consiste en que el análisis del proceso se realiza directamente en el protocolo del proceso, independientemente de que los datos almacenados en el protocolo del proceso describan procesos "simples" o "complejos", que pueden comprender subprocesos anidados y ejecutados en paralelo.

REIVINDICACIONES

1. Método implementado por ordenador para determinar trayectos de proceso paralelos en los datos del proceso en un sistema informático que tiene un procesador, en donde
 - los datos del proceso se almacenan en al menos un sistema fuente como parte de los datos fuente y comprenden al menos una instancia del proceso ejecutada,
 - al menos una instancia del proceso comprende una serie de etapas del proceso, en donde al menos dos etapas del proceso se ejecutan en paralelo como subprocesos de la instancia del proceso,
 - el procesador lee las etapas del proceso a partir de los datos fuente y los almacena en un dispositivo de almacenamiento del sistema informático de acuerdo con una estructura de datos predeterminada, en donde la estructura de datos predeterminada comprende al menos:
 - un primer atributo, en el que se almacena un identificador único para la instancia del proceso de la etapa del proceso correspondiente,
 - un segundo atributo, en el que se almacena un identificador único para la etapa del proceso correspondiente,
 - un tercer atributo, en el que se almacena la secuencia de las etapas del proceso dentro de una instancia del proceso, y
 - un identificador de jerarquía, en el que se almacena la jerarquía de las etapas del proceso,
 - en donde las etapas del proceso se ordenan de acuerdo con el tercer atributo, de modo que, en la estructura de datos, las etapas del proceso sucesivos de una instancia del proceso se almacenan en porciones de memoria inmediatamente adyacentes del dispositivo de almacenamiento, en donde el dispositivo de almacenamiento del sistema informático es una memoria principal que está acoplada operativamente al procesador, en donde el procesador del sistema informático determina o calcula, a partir de los datos fuente, el valor del identificador de jerarquía para cada etapa del proceso, y
 - en un primer etapa, se genera una jerarquía de estados a partir de las etapas del proceso almacenados en el dispositivo de almacenamiento, en donde las etapas del proceso se leen como un flujo del dispositivo de almacenamiento y se añaden secuencialmente a la jerarquía de estados, en donde se añade a la jerarquía de estados una relación de precedente y/o sucesor con respecto a al menos una etapa del proceso más para cada etapa del proceso añadido, y
 - en un segundo etapa, una jerarquía de procesamiento de la instancia del proceso que comprende nodos y bordes se deriva de las etapas del proceso almacenados en la jerarquía de estados y las respectivas relaciones precedentes y/o sucesores, en donde cada nodo corresponde a una etapa del proceso y cada borde corresponde a una relación de precedente/sucesor entre dos etapas del proceso, en donde al menos un nodo de jerarquía de procesamiento tiene al menos dos sucesores, que son etapas del proceso de los subprocesamientos ejecutados en paralelo.
2. Método de acuerdo con la reivindicación 1, en donde el identificador de jerarquía comprende
 - un cuarto atributo, en donde se almacena un identificador para un subprocesamiento de nivel superior para la etapa del proceso correspondiente, y
 - un quinto atributo, en el que se almacena un identificador único para el subprocesamiento al que pertenece la etapa del proceso correspondiente.
3. Método de acuerdo con la reivindicación 1, en donde, antes de la primera etapa, se clasifican físicamente las etapas del proceso de cada instancia del proceso en el dispositivo de almacenamiento, primero de acuerdo con el primer atributo y luego de acuerdo con el tercer atributo.
4. Método de acuerdo con la reivindicación 2, en donde, al añadir la etapa del proceso a la jerarquía de estados,
 - el valor del quinto atributo de la etapa del proceso de la jerarquía de estados se añade como identificador del subproceso correspondiente, siempre que dicho valor no se haya añadido aún a la jerarquía de estados, y
 - el valor del segundo o tercer atributo de la etapa del proceso en la jerarquía de estados se asigna como identificador de la etapa del proceso a dicho identificador para el subproceso.
5. Método de acuerdo con la reivindicación 3, en donde, tras la adición de la etapa del proceso a la jerarquía de estados, se determina el precedente de la etapa del proceso añadida y se añade la relación de precedente/sucesor a la jerarquía de estados de la etapa del proceso añadida y al precedente determinado.
6. Método de acuerdo con la reivindicación 4, en donde, al determinar el precedente de la etapa del proceso añadida, se comprueba si el valor del cuarto atributo es diferente al valor del quinto atributo para la etapa del proceso añadida, y
 - si el valor del cuarto atributo es diferente al valor del quinto atributo, se comprueba si ya se ha asignado un identificador de etapa del proceso al identificador del subproceso al añadir la etapa del proceso a la jerarquía de estados, y
 - si ya se ha asignado un identificador de etapa del proceso al identificador del subproceso, la etapa del proceso que tiene el identificador de la etapa del proceso ya asignado es el precedente de la etapa del proceso añadida,

- si aún no se ha asignado un identificador de la etapa del proceso al identificador del subproceso, la última etapa del proceso añadida al subproceso de nivel superior es el precedente de la etapa del proceso añadida, en donde el subprocesamiento de nivel superior se identifica por el cuarto atributo de la etapa del proceso añadida,
- si el valor del cuarto atributo no difiere del valor del quinto atributo, la última etapa del proceso añadida al subprocesamiento identificado por el cuarto atributo de la etapa del proceso añadida es el precedente de la etapa del proceso añadida.

5

7. Método de acuerdo con la reivindicación 1, en donde la jerarquía de estados se genera en la memoria principal del sistema informático.

10

8. Método de acuerdo con la reivindicación 1, en donde el identificador único de la etapa del proceso comprende una descripción de la etapa del proceso.

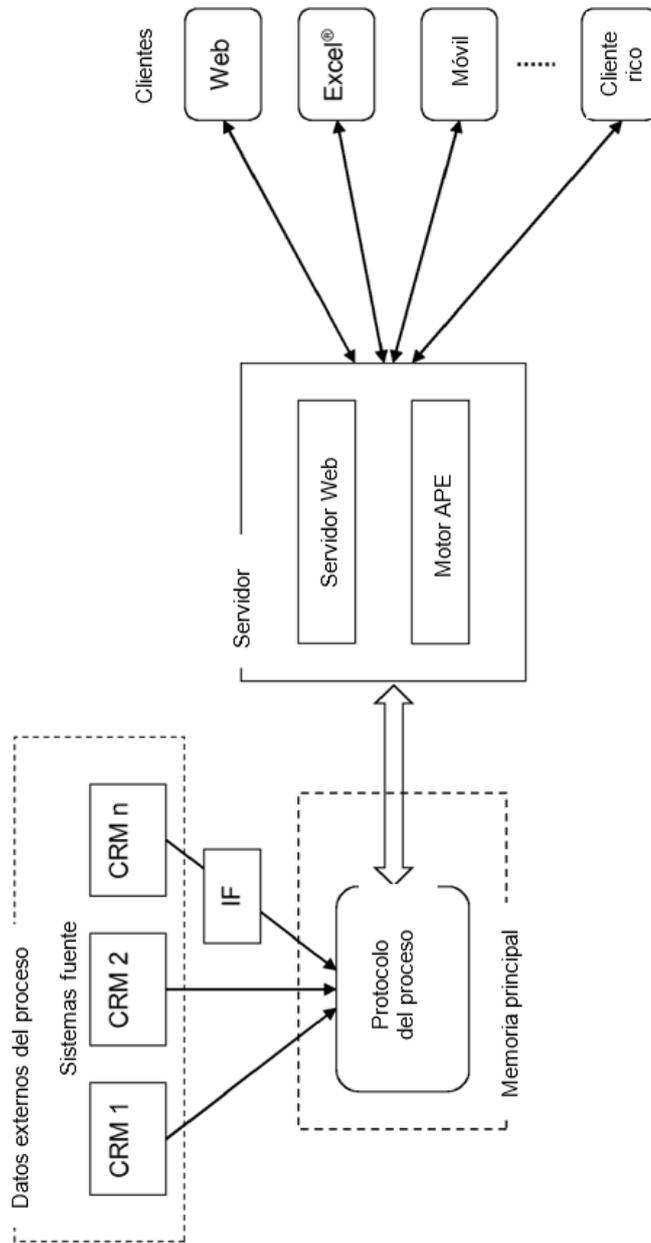


Fig. 1

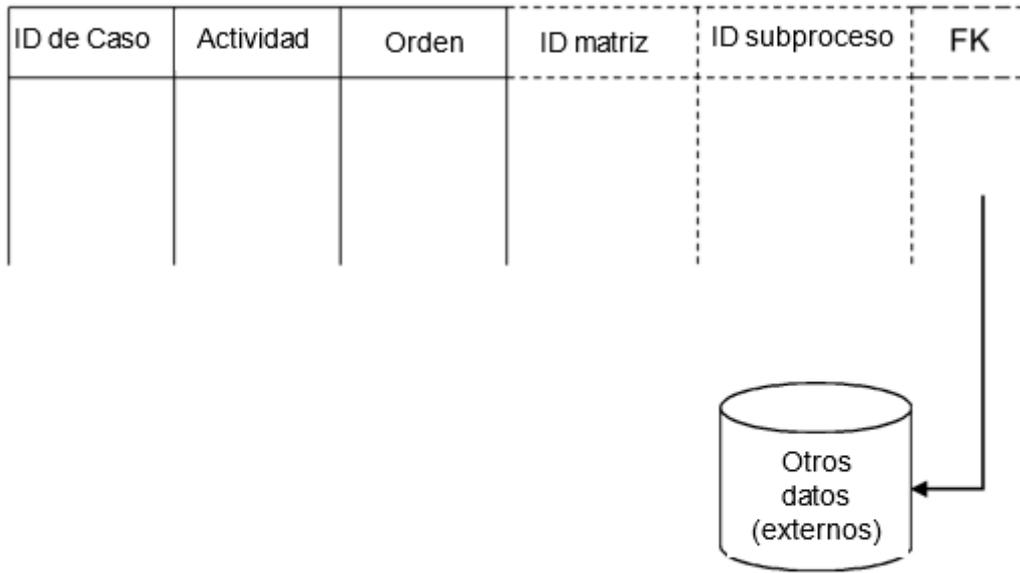


Fig. 2

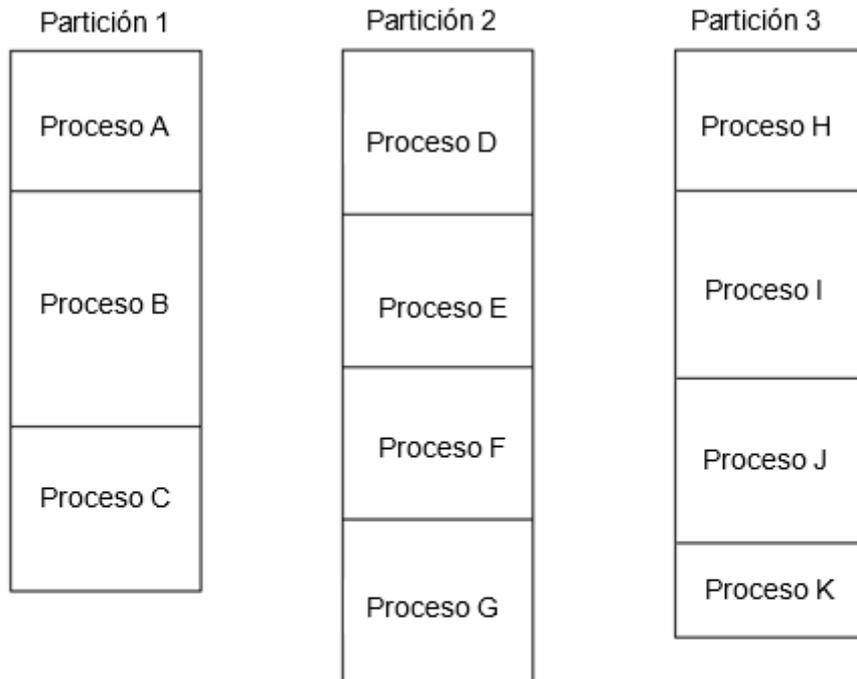


Fig. 2 (a)

ID Caso	Actividad	Orden	ID Matriz	ID Subproceso
1	A	1		Principal
1	B	2		Principal
1	C	3	Principal	SP1
1	D	4	Principal	SP2
1	E	5	Principal	SP1
1	F	6		Principal

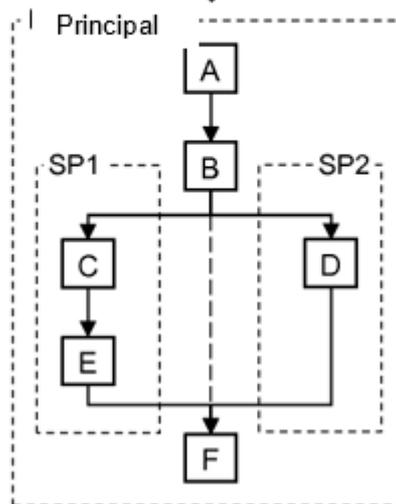
(a)



Jerarquía de estado

	Etapa 1	Etapa 2	Etapa 3	Etapa 4	Etapa 5	Etapa 6
Principal	1	2	2	2	2	6
SP1	1	2	3	3	5	6
SP2	1	2	2	4	4	6
bordes		1→2	2→3	2→4	3→5	2→6 5→6 4→6

(b)



(c)

Fig. 3

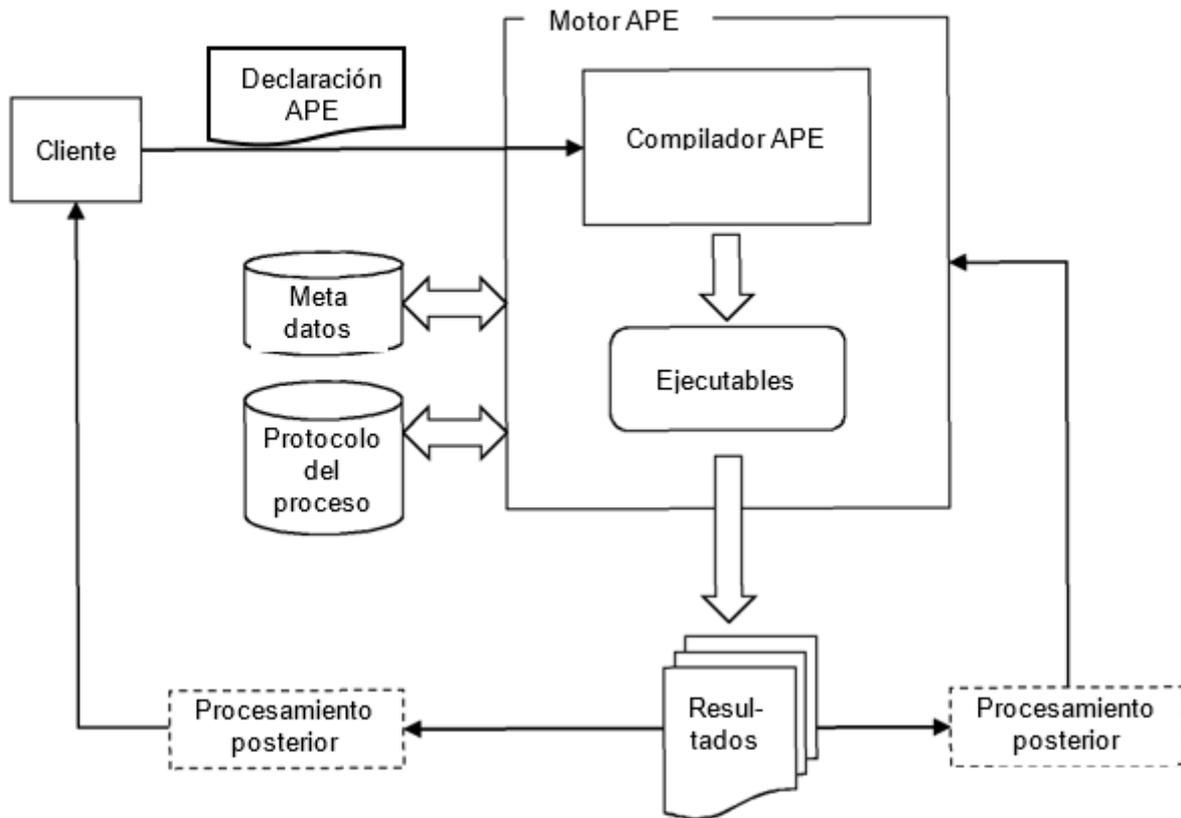


Fig. 4

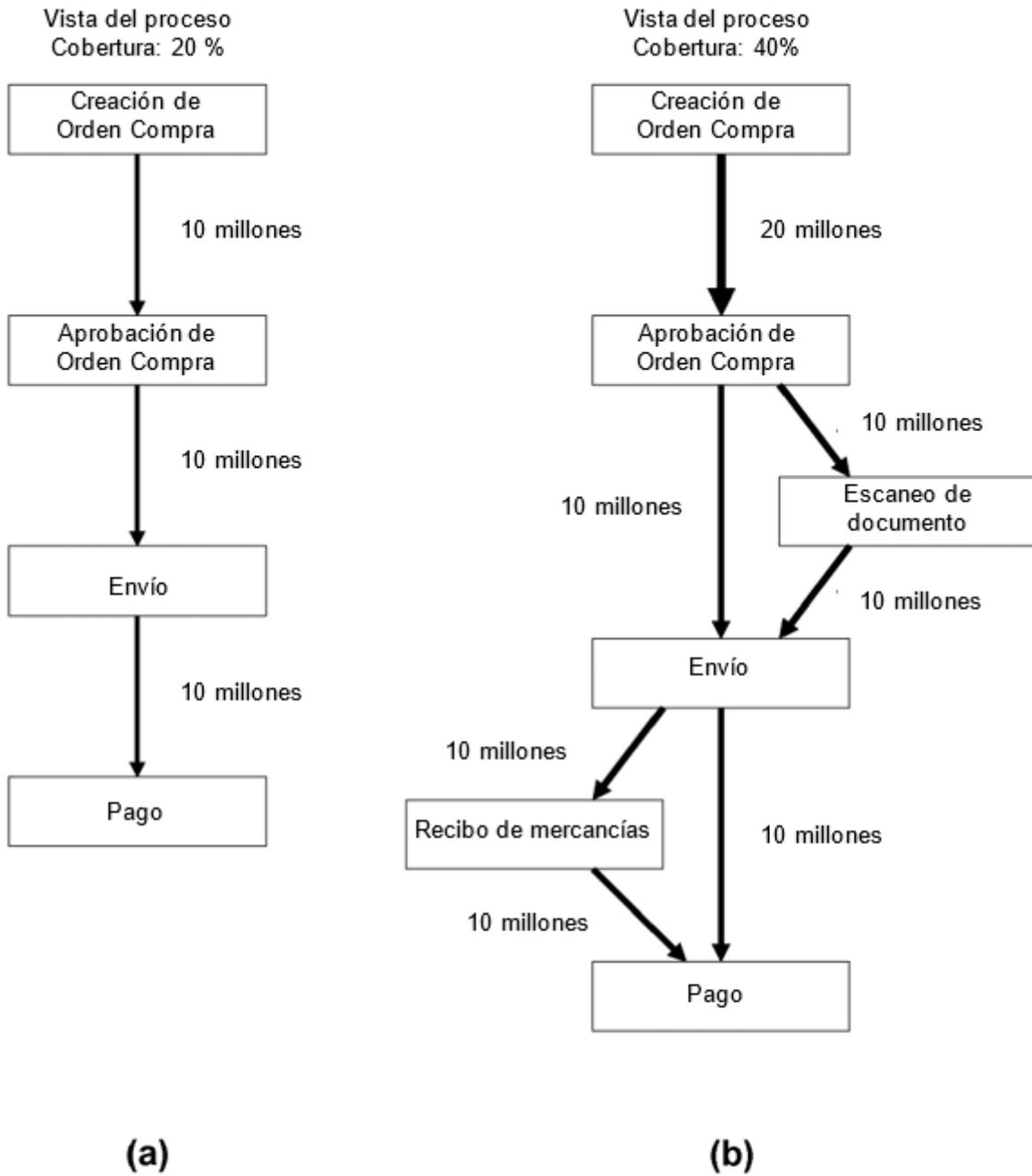


Fig. 5