

19



OFICINA ESPAÑOLA DE
PATENTES Y MARCAS

ESPAÑA



11 Número de publicación: **2 923 288**

51 Int. Cl.:

G06F 9/46 (2006.01)
H04L 29/08 (2006.01)
G05B 15/00 (2006.01)
H04L 12/28 (2006.01)
B25J 9/16 (2006.01)
G06F 9/54 (2006.01)

12

TRADUCCIÓN DE PATENTE EUROPEA

T3

86 Fecha de presentación y número de la solicitud internacional: **14.02.2019** **PCT/FR2019/050328**

87 Fecha y número de publicación internacional: **29.08.2019** **WO19162595**

96 Fecha de presentación y número de la solicitud europea: **14.02.2019** **E 19710051 (4)**

97 Fecha y número de publicación de la concesión europea: **13.04.2022** **EP 3756091**

54 Título: **Procedimiento de interconexión entre robots**

30 Prioridad:

20.02.2018 FR 1851429

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

26.09.2022

73 Titular/es:

**UNIVERSITÉ DE REIMS CHAMPAGNE-ARDENNE
(100.0%)
Villa Douce 9 Boulevard de la Paix
51100 Reims, FR**

72 Inventor/es:

**AYAIDA, MARWANE;
MESSAI, NADHIR;
VALENTIN, FRÉDÉRIC;
MARCHERAS, DIMITRI y
AFILAL, LISSAN**

74 Agente/Representante:

CURELL SUÑOL, S.L.P.

ES 2 923 288 T3

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

DESCRIPCIÓN

Procedimiento de interconexión entre robots

5 Campo y contexto

La presente invención se refiere al campo de la comunicación entre los objetos conectados (robots móviles, sensores, etc.), y de la interoperabilidad de estos objetos conectados.

10 Es conocido utilizar unos robots móviles de diferentes tipos, tales como unos robots humanoides, unos carros móviles con o sin brazo manipulador, unos drones, etc. Estos robots están asociados muy frecuentemente a diferentes sensores de adquisición de datos del entorno, y provistos de una comunicación *ad-hoc* y de sistemas operativos y de control y mando diferentes. La realización de estos diferentes objetos, más o menos autónomos, para hacer que ejecuten unas operaciones complejas, organizando una distribución de las tareas a realizar y una cooperación de estos objetos entre sí, necesita unos medios de comunicación y de intercambios de información entre unos equipos materiales y unos softwares frecuentemente muy heterogéneos.

Hasta la fecha, no existe ningún software o programa de interfaz, también denominado software intermedio, suficientemente genérico para asegurar la comunicación y la interoperabilidad deseadas entre estos objetos móviles (robots y/u otros sensores de datos), heterogéneos tanto por sus características físicas como por su software operativo respectivo, permitiendo en particular la apertura de servicios aplicativos de alto nivel, así como la integración fácil y unificada de bloques de software de terceros. El documento US 2006/079997 A1 describe un procedimiento de gestión de un conjunto de robots con el fin de establecer unas trayectorias. El documento US 2013/123980 A1 describe un procedimiento de gestión por unos operarios de un conjunto de robots que funcionan en cooperación. El documento US 6 374 155 B1 describe un procedimiento de gestión de un conjunto de robots que comprende por lo menos un robot navegador encargado de pilotar otras unidades.

El problema a resolver es, por lo tanto, diseñar un software de interfaz capaz de agrupar unos objetos móviles heterogéneos, permitiendo en particular interconectar estos objetos que pueden pertenecer a una o varias flotas de objetos para la realización de tareas específicas, eventualmente a distancia, entre sí y/o mediante uno o varios operarios.

La invención tiene, por lo tanto, como objetivo proponer una solución a la totalidad o parte de este problema.

35 Descripción de la invención

Con este fin, la presente invención se refiere a un procedimiento para la interconexión de un primer robot con por lo menos un segundo robot, y a un procedimiento para la gestión de la cooperación entre una pluralidad de robots, así como a por lo menos un programa de ordenador que comprende las instrucciones configuradas para ejecutar dichos procedimientos.

La presente invención se refiere a un procedimiento para la interconexión de un primer robot con por lo menos un segundo robot, comprendiendo cada robot por lo menos:

- 45 - una unidad de control y mando, comprendiendo la unidad de control y mando por lo menos un procesador y por lo menos un módulo de memoria;
- por lo menos un equipo periférico de tipo accionador o sensor;
- 50 - por lo menos un equipo de comunicación conectado a por lo menos una red de comunicaciones;
- un primer programa P1, configurado para pilotar el por lo menos un equipo de comunicación, y que comprende una pluralidad de primeros módulos de software que se ejecutan en el por lo menos un procesador;
- 55 - un segundo programa P2, que comprende una pluralidad de segundos módulos de software que se ejecutan en el por lo menos un procesador, estando la configuración de dicho segundo programa determinada por un archivo de parametrización que comprende un campo *typ_service* que indica un modo de envío entre los modos interpretar o directo, caracterizando dicho modo de envío los mensajes transmitidos por dicho segundo programa
- 60 - un tercer programa P3, configurado para mandar el por lo menos un accionador y/o el por lo menos un sensor, y que comprende una pluralidad de terceros módulos de software que se ejecutan en el por lo menos un procesador;

65 comprendiendo el procedimiento las etapas siguientes, realizadas por el segundo programa dentro de cada uno

de los robots, después de la recepción por parte del segundo programa de un primer mensaje procedente del tercer programa:

- conversión del primer mensaje en un segundo mensaje, siendo el segundo mensaje formateado según una estructura objeto predefinida que comprende un campo typ_msg que indica un tipo de dicho mensaje de entre los tipos: orden, solicitud, o información,
- transmisión del segundo mensaje a por lo menos un programa de entre: primer programa perteneciente al mismo robot, segundo programa de otro robot;

comprendiendo además el procedimiento de interconexión las etapas siguientes, realizadas por el segundo programa dentro de cada uno de los robots, después de la recepción por parte del segundo programa de un tercer mensaje, formateado según dicha estructura objeto predefinida, procedente de por lo menos un programa de entre: primer programa perteneciente al mismo robot, segundo programa de otro robot, cuando el campo typ_msg de dicho tercer mensaje indica el tipo orden o el tipo solicitud:

- traducción del tercer mensaje en una parte de programa apta para ser realizada por el tercer programa cuando el campo typ_service del archivo de parametrización asociado a dicho tercer mensaje indica el modo de envío directo; y/o
- transmisión según un segundo protocolo de transporte, de por lo menos una parte de dicho tercer mensaje, o de dicha parte de programa, con destino al tercer programa.

Ventajosamente, la estructura objeto predefinida y genérica, a través de la cual el segundo programa P2 asegura la configuración de las informaciones y/o instrucciones a transmitir, está compuesta por los campos siguientes:

- tipo de mensaje: orden, información o solicitud;
- subtipo de mensaje: tipo de datos (posición, real, imagen, vídeo, etc.) o de orden (avanzar, retroceder, parar, etc.);
- fuente: dirección del expedidor, compuesta por el par dirección IP/Puerto o por el nombre de la fuente (pudiendo el par dirección IP/Puerto estar asociado a la tabla de enrutamiento);
- destino: dirección de destinatario, compuesta por el par dirección IP/Puerto, o nombre del destinatario;
- tamaño del campo de datos;
- datos.

Según un aspecto de la invención, la etapa de traducción comprende el acceso dentro del segundo programa a un archivo de configuración (que pertenece a P2) tal como una tabla de correspondencia o una base de datos registrada en el por lo menos un módulo de memoria, proporcionando dicho archivo de configuración al segundo programa la parte de programa apta para ser realizada por el tercer programa.

Preferentemente, dicho archivo de configuración permite que el segundo programa:

- realice una correspondencia entre el subtipo del mensaje (definido en el campo subtipo de la estructura objeto presentada anteriormente) y los datos (definidos en el campo de datos de la estructura objeto presentada anteriormente)
- seleccione la parte de programa apta para ser realizada por el tercer programa (traducción de la orden o de la solicitud al lenguaje específico del robot para que este último pueda ejecutarlo directamente).

Según un aspecto de la invención, la etapa de transmisión de dicha por lo menos una parte de dicho tercer mensaje, o de dicha parte de programa, comprende por lo menos:

- una primera transmisión, según un segundo protocolo de transporte, de un cuarto mensaje del segundo programa hacia un cuarto programa,
- una segunda transmisión de un quinto mensaje de dicho cuarto programa con destino al tercer programa.

Según un aspecto de la invención, se utiliza un primer protocolo de transporte para la transmisión de un sexto mensaje entre el segundo programa del primer robot y el segundo programa del segundo robot.

Según un aspecto de la invención, el primer, segundo, tercero, cuarto, quinto y sexto mensajes intercambiados

están en formato JSON.

Según un aspecto de la invención, el primer protocolo es del tipo UDP.

5 Según un aspecto de la invención, el segundo protocolo es del tipo TCP.

Según un aspecto de la invención, el por lo menos un equipo de telecomunicaciones comprende un medio de comunicación inalámbrica.

10 Según un aspecto de la invención, el medio de comunicación inalámbrica comprende por lo menos un módulo de entre: módulo de tipo WIFI, módulo de tipo RFID, módulo de tipo Bluetooth, módulo de tipo GSM.

Según un aspecto de la invención, el segundo programa del primer robot cifra el segundo mensaje antes de transmitirlo si el destinatario es el por lo menos un segundo robot, y el segundo programa del primer robot descifra el tercer mensaje antes de traducirlo o transmitirlo, si el expedidor es el por lo menos un segundo robot.

15 Así, dentro de cada robot, el segundo programa es apto para cifrar y/o descifrar inteligentemente los mensajes según su procedencia y su destino.

20 Según un aspecto de la invención, el por lo menos un segundo robot comprende un operario remoto conectado a un servidor internet.

Según un aspecto de la invención, el tercer programa comprende el sistema operativo del robot en el que está instalado, y un programa piloto de por lo menos un accionador y/o de por lo menos un sensor.

25 La invención se refiere asimismo a un procedimiento para la gestión de la cooperación entre una pluralidad de robots, comprendiendo cada robot de la pluralidad de robots por lo menos:

- 30 - una unidad de control y mando, comprendiendo la unidad de control y mando por lo menos un procesador y por lo menos un módulo de memoria;
- por lo menos un equipo periférico de tipo accionador o sensor;
- 35 - por lo menos un equipo de comunicación conectado a por lo menos una red de comunicaciones;
- un primer programa, configurado para pilotar el por lo menos un equipo de comunicación, y que comprende una pluralidad de primeros módulos de software que se ejecutan en el por lo menos un procesador;
- 40 - un segundo programa, que comprende una pluralidad de segundos módulos de software que se ejecutan en el por lo menos un procesador,
- un tercer programa, configurado para mandar el por lo menos un accionador y/o el por lo menos un sensor, y que comprende una pluralidad de terceros módulos de software que se ejecutan en el por lo menos un procesador;

45 comprendiendo el procedimiento por lo menos las etapas siguientes:

- 50 - emisión por una capa aplicativa alojada en un servidor o accesible a través de internet, de un archivo de misión, comprendiendo dicho archivo de misión una lista de tareas elementales planificadas y su secuenciación, y de un archivo de identificación y de priorización de los equipos periféricos,
- recepción de dicho archivo de misión y de dicho archivo de identificación y de priorización, por un servicio de equipo del primer programa de cada robot de la pluralidad de robots,
- 55 - transmisión, por parte del primer programa de cada robot de la pluralidad de robots, de un primer mensaje de tipo solicitud con destino a su segundo programa, para conocer un identificador de cada periférico disponible del por lo menos un periférico de cada robot de la pluralidad de robots,
- 60 - realización, por parte del segundo programa de cada robot de la pluralidad de robot, del procedimiento para la interconexión de robots según la invención para solicitar al tercer programa (dentro del mismo robot) y transmitir al primer programa (dentro del mismo robot) los identificadores de los periféricos disponibles en cada robot de la pluralidad de robots;
- 65 - tratamiento, por parte del servicio de equipo del primer programa de cada robot de la pluralidad de robots, de los identificadores de los periféricos disponibles para generar una lista de las tareas realizables por cada robot de la pluralidad de robots (cada robot determina en su interior la lista de tareas que puede realizar en

función de los periféricos de los que dispone),

- transmisión a cada robot de la pluralidad de robots y/o a un servicio centralizado del servidor, por cada servicio de negociación/elección del primer programa, de dicha lista de las tareas realizables por cada robot de la pluralidad de robots,
- producción, por un servicio centralizado del servidor (nube) y/o tras un proceso de comunicación entre cada servicio de negociación/elección de los robots, en base al archivo de identificación y de priorización de los periféricos y de la lista de las tareas realizables, de una lista de los robots que participan en la misión,
- ejecución de la misión,
- sincronización de los robots,
- supervisión de la ejecución de la misión.

La invención se refiere asimismo a un programa de ordenador que comprende un conjunto de instrucciones configuradas para ejecutar por lo menos un procedimiento según la invención.

Preferentemente, la invención se refiere a un programa de ordenador que comprende un conjunto de instrucciones configuradas para ejecutar el procedimiento para la interconexión de robots, y a un programa de ordenador que comprende un conjunto de instrucciones configuradas para ejecutar el procedimiento de gestión de cooperación entre robots.

Figuras

Para su buena comprensión, la invención se describe con referencia a los dibujos adjuntos que representan, a título de ejemplo no limitativo, una forma de realización de un dispositivo según la invención.

La figura 1 es una representación esquemática de un robot móvil según una primera variante de la invención totalmente integrada en el mismo, y de su entorno.

La figura 2 es una representación esquemática de una flota de robots comunicantes según dicha primera variante de la invención totalmente integrada en estos.

La figura 3 es una representación esquemática de la arquitectura de la invención según una segunda variante, en la que la invención está, por un lado, parcialmente integrada en los robots y, por otro lado, parcialmente trasladada hacia la nube.

La figura 4 es una representación esquemática de los diferentes actores (robots, usuarios) y de las diferentes vías de comunicaciones posibles según la invención.

La figura 5 ilustra un diagrama de comunicación entre un primer robot R1 y un segundo robot R3.

La figura 6 ilustra un diagrama de comunicación entre un usuario U1 y un robot R3.

Las figuras 7 y 8 presentan esquemáticamente un escenario de colaboración de robots según la invención.

Descripción detallada

La invención, y sus modos de realización, se describen con referencia a los dibujos adjuntos y representados en las figuras 1 a 8.

La invención se refiere a un procedimiento que es realizado por un segundo programa P2, en un robot autónomo 1 tal como el representado esquemáticamente en la figura 1; P2 comprende uno o varios subprogramas o módulos de software, y constituye una capa de software que asegura la interfaz entre un primer programa P1 y otros dos programas, un tercer programa P3 y un cuarto programa P4; estos programas P1, P3 y P4 comprenden cada uno asimismo uno o varios subprogramas o módulos de software, y constituyen cada uno una capa de software.

El robot 1, tal como se presenta en la figura 1, está en comunicación con:

- otros robots 2, a los que puede proporcionar unas informaciones y/o unas órdenes y a través de los cuales puede recibir unas informaciones y/o unas órdenes, de forma pasiva y/o bajo demanda en ambos casos, y
- con un servidor remoto 3 por medio del cual:

- un operario puede especificar a distancia las tareas a realizar por el o los robots que interactúan entre sí,
- el robot 1 puede comunicarse (informaciones y/u órdenes, de forma pasiva y/o bajo demanda) con los otros robots 2.

Según una variante de la invención, los robots 1 y 2 son aptos para comunicarse directa y exclusivamente entre sí (por ejemplo, en un modo de funcionamiento degradado en el que el servidor remoto 3 está averiado, o, por ejemplo, cuando se desea realizar mantenimiento y/o por razones de costes prescindiendo del uso de un servidor remoto 3).

- El robot autónomo 1 que sirve en este caso para ilustrar la descripción del procedimiento puede ser un robot autónomo o cualquier otro objeto conectado siempre que este objeto comprenda los principales equipos materiales representados en la figura 1: por lo menos un equipo periférico de tipo accionador A1, A2, para la movilidad o la prensión de carga, etc. y/o de tipo sensor C para la adquisición de datos del entorno (posición geográfica, temperaturas, etc.);
- por lo menos una unidad de control y mando del o de los equipos periféricos, que comprende por lo menos un procesador 4 y por lo menos un módulo de memoria M; los programas o capas de software P1, P2, P3 y P4 (módulo de adaptación) se ejecutan en el procesador 4 de la unidad de control y mando (se puede observar asimismo en la figura 2).
- por lo menos un equipo de telecomunicaciones T1, T2, tales como, por ejemplo, unos módulos de comunicación WIFI, o Bluetooth, o GSM, que permiten que el objeto considerado 1 se comunique con un servidor remoto 3 y/o con otro objeto conectado 2;

El primer programa P1 asegura la gestión de los medios de telecomunicación T1, T2 y la atención de los protocolos de transporte de los mensajes que se intercambian entre los subprogramas de un mismo robot u objeto conectado, y entre los programas y subprogramas que pertenecen a dos robots u objetos conectados diferentes. Ventajosamente, el primer programa P1 comprende un conjunto de módulos de software tales como (no limitativo):

- un servicio de colaboración,
- un servicio de negociación/elección,
- un servicio de equipos,
- un bus de comunicación que comprende, entre otros, un servicio de detección de periféricos, un servicio de sincronización de datos.

El segundo programa P2, o segundo conjunto de subprogramas o módulos de software realiza el procedimiento según la invención.

Los primer y segundo programas P1, P2 son genéricos y se pueden abrir en cualquier robot, independientemente del sistema operativo.

El tercer programa P3, o tercer conjunto de subprogramas o módulos de software, comprende el sistema operativo específico de cada robot u objeto conectado, y los controladores específicos de los accionadores y/o sensores asociados al robot o al objeto conectado considerado. Ventajosamente, el programa P3 comprende un conjunto de módulos de software tales como (no limitativo):

- sistema operativo robótico (por ejemplo: ROS, Universal Robot OS, ABB OS): bibliotecas nativas, paquete de navegación, etc.
- framework nativo: controladores, bibliotecas, etc.

Mediante la presente descripción, en particular el tercer programa P3, se entiende claramente que la invención se articula alrededor del OS robótico específico del robot con el fin de proporcionar y garantizar su interoperabilidad dentro de una o varias flotas de robots.

El cuarto programa P4, o cuarto conjunto de subprogramas o módulo de software de adaptación, es el único módulo de software de la invención que es necesario adaptar al robot en el que se instala la invención.

Una biblioteca de adaptación, externa al robot y a los programas P1, P2, P3, P4, permite generar en particular el esqueleto de dicho cuarto programa en el lenguaje propio del robot.

A continuación, en cursiva en lenguaje de programación (los comentarios están en negrita), se presenta un ejemplo del esqueleto del cuarto programa P4 tal como se genera a través de dicha biblioteca de adaptación, y que debe ser completado y finalizado por el operario según sus necesidades (entre otros a nivel de las funciones "dothis", "dothat"):

////////

función que permite conectarse a P2

open=socket_open('ip','port')

ip, port son, por lo tanto, la dirección del robot o del ordenador (local, nube) en el que se encuentra P2

while (open == False):

open=socket_open('ip','port')

end

////////

función que puede utilizarse para recibir un mensaje por parte de P2 (campo por campo (modo "interpretado"))

def recv_msg():

socket_send_string('ready')

notifica P2 que el robot está dispuesto a recibir un mensaje

while from=="":

from = socket_read_string()

se bloquea hasta recibir un primer campo

end

type=socket_read_ascii_float(1)

y después se reciben los otros campos uno por uno

stype=socket_read_string()

size=socket_read_ascii_float(1)

data=socket_read_string()

end

//////////

función que permite enviar un mensaje hacia P2 (que se encargará de transmitirlo al objetivo indicado en el mensaje)

def send_msg(message):

socket_send_string(message)

end

////////////////////////////////

Función en la que el operario define el comportamiento a adoptar según el valor de los diferentes campos del mensaje recibido. Las funciones "dothis", "dothat" serán, por lo tanto, generalmente, una instrucción o un conjunto de instrucciones propias del robot (lo cual caracteriza la naturaleza del cuarto programa, único módulo de software de la invención que es necesario adaptar al robot)

```
def handle():
    if type=="commande":
        dothis
    if type=="request"
        dothat
    ///////
```

La biblioteca de adaptación está disponible, por ejemplo, en la nube, o disponible a través de un sistema que se puede conectar al robot bajo demanda, tal como un ordenador local o una tableta.

La biblioteca de adaptación permite además asistir al operario en la definición y la configuración del o de los escenarios de interacción entre los robots.

El módulo de software de adaptación (cuarto programa P4) se personaliza así en función de las necesidades del operario, y se genera por lo menos parcialmente a través de dicha biblioteca de adaptación. Comprende, entre otras, cuatro funciones:

- "Conexión": esta función permite conectar el robot a las capas de software P1 y P2, se basa en particular en la indicación de un par Dirección IP/Puerto correcto.
- "Envío de mensaje": un mensaje está compuesto por varios campos enviados los unos tras los otros, preferentemente estos mensajes están compuestos por seis campos (tipo, subtipo, fuente, destino, tamaño, datos).
- "Recepción de mensaje": los datos recibidos deben respetar una estructura específica tal como se detalla a continuación; por otro lado, en cuanto un robot es capaz de recibir datos, el programa P4 envía un mensaje a la capa P2 notificándole su capacidad para recibir mensajes.
- "Tratamiento de mensaje": esta función está adaptada específicamente a cada situación, define en el lenguaje propio del robot los comportamientos a adoptar en función, por un lado, de los mensajes y datos recibidos y, por otro lado, de la situación. En particular, esta función define la manera en la que el robot transmite en su interior una orden, una solicitud, una información (comportamiento comunicativo del robot) desde el cuarto programa P4 hacia el tercer programa P3.

En cada uno de los robots u objetos interconectados, la capa de software P1 está configurada para transportar los mensajes, debidamente formateados previamente, de un robot (u objeto) a otro robot (u objeto) y/o al servidor remoto 3, a través de los equipos de telecomunicaciones disponibles para los intercambios entre robots y/u objetos remotos. Juntos, las capas de software P1 y los equipos de telecomunicaciones utilizados por estas capas de software aseguran, entre otras, la función de un bus de comunicación BC, material y software, que permite conectar los diferentes robots entre sí y a la nube. Preferentemente, cada bus de comunicación BC dispone de un enrutador R que les es propio, que permite, entre otros, unir el robot directamente a la nube. Las capas de software superiores alojadas por el servidor, por ejemplo laZ IHM, o en la nube, se denominan capas aplicativas.

El programa intermedio o la capa de software intermedio, denominada frecuentemente "software intermedio" o "middleware", P2 está configurado para:

- por un lado, transformar las informaciones u órdenes recibidas a través del bus de comunicación del nivel superior P1 y/o procedentes de los otros robots (en particular procedentes del segundo programa del robot expedidor) y/o del servidor remoto 3, en informaciones o instrucciones comprensibles o ejecutables por la capa de software inferior P3 que es específica del robot considerado para el pilotaje de los accionadores y/o sensores;
- por otro lado, en sentido inverso, formatear de una manera genérica los mensajes de informaciones o de órdenes destinadas a ser transmitidas por el robot considerado hacia los otros robots (u objetos conectados) y/o el servidor remoto 3; los mensajes así formateados genéricamente serán recibidos y después transformados por la capa de software intermedio P2 del robot o del objeto conectado destinatario.

A diferencia de las capas de software P3 y P4 de cada robot u objeto conectado, que son heterogéneas y específicas de cada robot u objeto, la capa de software intermedio P2 es genérica. Esta capa de software intermedio P2 asegura la interfaz entre los formatos e instrucciones propietarios realizados en cada robot por la capa de software P3 para el pilotaje de los accionadores y actores del robot considerado, y un formato genérico común al conjunto de los robots y objetos conectados; esta función de interfaz comprende en particular un mecanismo de traducción de las informaciones u órdenes recibidas según el formato genérico en informaciones o

instrucciones ejecutables según una de las capas de software P4 o P3 del robot destinatario.

Tal como se ilustra en la figura 2, los otros robots 2, 21, con los que se comunica el robot 1 pueden comprender:

- 5 • unos equipos materiales (accionador y/o sensor, unidad de control y mando con procesador y módulo de memoria, equipo de telecomunicación) de la misma naturaleza que los del robot 1,
- una arquitectura de software (programas P1, P2) de la misma naturaleza que la del robot 1,
- 10 • una arquitectura de software (programas P3, P4) que les es específica.

Por ejemplo, el procedimiento según la invención puede permitir interconectar el robot 1 que comprende las capas P1, P2, P31 y P41 con:

- 15 • un primer robot 2 que tiene un accionador, dos sensores, un equipo de telecomunicación de tipo WIFI, una unidad de control y mando de los equipos periféricos que comprende por lo menos un procesador y por lo menos un módulo de memoria; los programas o capas de software P1, P2, P32, P42 de dicho primer robot 2 se ejecutan en su procesador,
- 20 • un segundo robot 21 que tiene seis sensores, dos equipos de telecomunicación de tipo WIFI y Bluetooth, una unidad de control y mando de los equipos periféricos que comprende por lo menos un procesador y por lo menos un módulo de memoria; los programas o capas de software P1, P2, P33, P43 de dicho segundo robot 21 se ejecutan en su procesador.

25 Los programas P1, P2 de cada uno de los robots interconectados tienen los mismos bloques de software (idénticos).

30 La figura 3 ilustra un modo de realización de la invención, en el que los robots (en el sentido material) están provistos únicamente de los programas P3 y P4 que les son específicos, mientras que los programas genéricos comunes P1 y P2 son trasladados a la nube. Los robots se comunican entre sí por medio de un bus de comunicación BC material y de software, y de por lo menos un enrutador R al cual está asociada una tabla de enrutamiento TR, y eventualmente con la nube C y/o por lo menos un usuario humano a través de una interfaz hombre-máquina IHM.

35 Preferentemente, dicha tabla de enrutamiento comprende una lista de dobletes en la forma (nombre del robot, dirección IP: Puerto) que permite que el enrutador integrado ventajosamente en el segundo programa P2 redirija correctamente los mensajes. Así, un robot únicamente debe conocer el nombre del robot al que quiere enviar un mensaje.

40 Cuando llegan unas informaciones al segundo programa P2 para ser transmitidas a un robot u objeto destinatario, el segundo programa P2 asegura el conformado de las informaciones y/o instrucciones a transmitir en un mensaje según una estructura objeto predefinida y genérica, compuesta por los campos siguientes:

- 45 - tipo de mensaje: orden, información o solicitud;
- subtipo de mensaje: tipo de datos (posición, real, imagen, vídeo, etc.) o de orden (avanzar, retroceder, parar, etc.);
- 50 - fuente: dirección del expedidor, compuesta por el par dirección IP/Puerto o por el nombre de la fuente (pudiendo el par dirección IP/Puerto estar asociada con la tala de enrutamiento);
- destino: dirección del destinatario, compuesta por el par dirección IP/Puerto, o nombre del destinatario;
- tamaño del campo de datos;
- 55 - datos.

60 Esta estructura genérica permite una transmisión transparente de los mensajes en los niveles superiores (P1 y P2).

 La configuración de la capa de software P2 se basa en un formato de archivo JSON. Este archivo de parametrage define un cierto número de parámetros, que incluyen:

- 65 - dirección IP del robot (vista del servidor) en la que se ejecuta el segundo programa P2;

- nombre utilizado para identificar el robot;
- puerto de escucha externo, en el que el segundo programa P2 debe escuchar los mensajes externos,
- puerto de escucha interno, en el que debe conectarse el tercer o cuarto programa P3, P4 del robot;
- tipo de servicio, que permite indicar si los mensajes emitidos con destino al robot (con el que se desea comunicar) están destinados a ser "interpretados" (después de la recepción) por el robot destinatario, o si los mensajes se entregan según un modo directo.

Se entiende por "interpretados" el hecho de que el robot deba definir su comportamiento analizando los valores de los campos del mensaje recibido. Esta etapa de interpretación es realizada por el cuarto programa P4 y se preconfigura a través de la biblioteca de adaptación presentada anteriormente.

En servicio directo, si el tipo de mensaje es una orden o una solicitud, y gracias al archivo de configuración, el segundo programa P2 que se ejecuta en el robot destinatario traduce la orden o la solicitud en una orden directamente ejecutable en este robot destinatario.

Según un aspecto de la invención, el archivo de configuración se genera (por ejemplo, mediante un programa externo) a partir de la biblioteca de adaptación, preferentemente antes de su inyección en el segundo programa P2.

Preferentemente, el recurso en modo directo al archivo de configuración aparece únicamente cuando los mensajes son unas órdenes y/o unas solicitudes con destino a robots que permiten la ejecución de órdenes mediante un sistema externo.

A título de ejemplo, se presenta a continuación en cursiva, la estructura en lenguaje de programación de un archivo de configuración, que hace corresponder en particular a la orden "move" la parte de programa traducida por

"/cmd_vel" en lenguaje ROS o la parte de programa "movej" en lenguaje UR (Universal Robot):

```
{
  "commands":
  {
    "move":
    {
      "ros": "/cmd_vel",
      "ur": "movej"
    },
    "despegar": "/ardrone/takeoff",
    "aterrizar": "/ardrone/land"
  },
  "requests"
  {
    "modal": "mandatory, needed for streaming, just modify the topic name if necessary",
    "modal": "/pose",

    "position": "/pose"
  }
}
```

La comunicación entre el segundo programa P2 y los módulos de las capas de software inferiores P3, P4 del robot considerado se realiza a través de una conexión de tipo TCP, representada en la figura 3 por las flechas continuas 30. La comunicación entre diferentes segundos programas P2 que pertenecen a una pluralidad de robots 1, 2, 21 se efectúa según un protocolo de tipo UDP representado en la figura 3 por las flechas discontinuas 40. Asimismo, la comunicación hacia el servidor y/o la nube C se efectúa según un protocolo de tipo UDP representado en la figura 3 por las flechas 41.

El funcionamiento global de la invención se recoge en la figura 4, con el fin de ilustrar otra variante de comunicación

en la que unas flotas diferentes de robots 50, 60 se comunican entre sí a través del procedimiento según la invención. Estas flotas pueden comprender además unos robots maestros 51, 61, aptos para transmitir los mensajes recibidos aguas arriba a través del bus de comunicación BC desde, por ejemplo, la otra flota, un usuario directo 70, la nube C, o por lo menos un usuario deslocalizado 80 que pasa por la nube C.

Por otro lado, preferentemente, cuando un nuevo robot desea unirse a una flota o entrar en comunicación con por lo menos otro robot que no lo reconoce aún, los primeros intercambios se efectúan según el protocolo UDP, y aún más preferentemente según el modo de difusión "broadcast".

El procedimiento de interconexión según la invención permite, gracias a un conjunto de capas y módulos de software muy mayoritariamente genérico, por un lado, interconectar los robots heterogéneos de una misma flota, pero también interconectar varias flotas entre sí.

Las órdenes se formulan a partir de la o de las capas de aplicación existentes (IHM remotas, IHM local en el sitio robótico o en el propio robot, nube, etc.).

Así, la invención se refiere asimismo a un procedimiento para la gestión de la cooperación entre robots heterogéneos (que presentan unos sistemas operativos diferentes), que comprende, entre otras, las etapas que consisten en:

- emitir desde la capa de aplicación (IHM, nube) un conjunto de archivos que presentan una misión, que comprende, entre otros, una lista de identificadores de tareas elementales planificadas (LTP), y un archivo de priorización de los periféricos,
- recibir a nivel del servicio "equipo" de la capa P1 de cada robot dicha lista de identificadores de tareas elementales planificadas (LTP),
- transmitir (dentro de un mismo robot) a partir de la capa P1 un mensaje de tipo solicitud con destino a la capa P2, que prevé conocer los periféricos disponibles en este robot,
- realizar el procedimiento de interconexión según la invención a nivel de la capa P2 con el fin de:
 - transmitir la solicitud a las capas P3 y/o P4 de manera adecuada,
 - recuperar las informaciones de periféricos requeridas procedentes de las capas P3 y/o P4,
 - transmitir a la capa P1 dichas informaciones, convertidas según la estructura objeto predefinida,
- tratar a nivel del servicio "equipo" las informaciones recibidas a nivel de la capa P1 y generar una lista de tareas realizables (LTR) por el robot,
- transmitir al conjunto de los robots, a partir del servicio "negociación/elección" de la capa P1, dicha lista de tareas realizables (LTR) por el robot
- intercambiar unos mensajes de negociación entre los diferentes servicios "negociación/elección" de las diferentes capas P1 de los diferentes robots,
- generar, a nivel de la nube y/o entre los robots en base a los mensajes de negociación intercambiados y al archivo de priorización de los periféricos, una lista (archivo) de los robots que participan en la misión (LRM),
- iniciar la ejecución del plan de desarrollo de la misión,
- sincronizar los robots,
- supervisar la colaboración entre los robots,

Preferentemente, el conjunto de archivos que formalizan una misión comprende por lo menos: una lista de identificadores de tareas elementales a ejecutar (LTP), un archivo de parámetros relativos a las limitaciones en la secuenciación de dichas tareas, una lista de los identificadores de los periféricos (accionadores, sensores, etc.) necesarios para la realización de las tareas, un archivo de priorización de los periféricos.

Preferentemente, la transmisión desde y con destino a la capa P1 se efectúa a través de un servicio de detección de periféricos que pertenece al bus de comunicación BC.

Preferentemente, la lista de tareas realizables (LTR) por el robot tiene forma de un archivo o forma de una base de datos que contiene, entre otros, el identificador del robot y la lista de los identificadores de tareas que puede realizar de entre la lista de identificadores de tareas elementales (LTP).

Preferentemente, el conjunto de los archivos intercambiados está en formato JSON.

5 La ejecución de la misión se puede iniciar automáticamente a través de los programas en la nube, o a través de los usuarios desde unas interfaces específicas IHM, remotas o locales.

Ventajosamente, los datos de periféricos se recuperan por un servicio de "sincronización de datos" que pertenece al bus de comunicación BC de la capa P1.

10 Ventajosamente, la supervisión de la colaboración entre robots se efectúa a nivel del servicio "colaboración" de la capa P1 de cada robot.

15 El diagrama de la figura 5 ilustra una comunicación entre un primer robot R1 y un segundo robot R3. Se describirán ahora las diferentes etapas de esta comunicación con, en cursiva, las instrucciones correspondientes del lenguaje de programación:

- Un mensaje $M(R1)_{P3 \rightarrow P2}$ se envía desde el tercer programa P3 del robot R1 al segundo programa P2 del robot R1:

20 $M(R1)_{P3 \rightarrow P2} = \text{"info/position/robot3/3.0.1"}$.

- Se efectúa entonces un tratamiento $T(R1)_{P2}$ dentro del robot R1 por parte de la capa P2, para formatear y/o reconstituir REC el mensaje según la estructura objeto predefinida:

Type = info

Sous-type = position

Target = robot3

Source = robot1

Size = 5

25 *Data = "3,0,1"*

- Seguido de una etapa denominada de serialización SE del mensaje por parte de la capa P2, por ejemplo en formato JSON, permitiendo este formato en particular asegurar la interoperabilidad y la facilidad para almacenar las informaciones *a posteriori*:

30 *{ "py/object" : "message.message ",*
"src" : {py/tuple" : ["10.12.1.130" , 48000"]},
"protocol" : 0,
"target" : {py/tuple" : [10.12.1.131, 48000"]},
"type_msg " : "info",
"sous_type" : "position",
"size" : 5,
"data" : "3,0,1"}

- Seguida de una etapa de firma por parte de la capa P2 (por ejemplo, por firma cifrada a partir de una clave interna dentro del código) del mensaje SI, asegurando esta etapa la integridad de la cadena JSON:

35 *087aa394104a8b6d9a3c06d35c768a0bcefad9*
{ "py/object" : "message.message ",
"src" : {py/tuple" : ["10.12.1.130" , 48000"]},
"protocol" : 0,
"target" : {py/tuple" : [10.12.1.131, 48000"]},
"type_msg " : "info",
"sous_type" : "position",
"size" : 5,

"data": "3,0,1"

- Se envía un mensaje $M(R1>R3)$ desde el robot R1 al robot R3.

- 5
- Se efectúa entonces un tratamiento $T(R3)_{P2}$ dentro del segundo programa del robot R3, para verificar VSI la firma y reconstituir REC el mensaje, que es transmitido después $M(R3)_{P2>P3}$ por el segundo programa P2 del robot R3 al tercer programa P3 del robot R3.

10 Por último, el diagrama de la figura 6 ilustra una comunicación entre un usuario U1 y un robot R3. Se describirán las diferentes etapas de esta comunicación con, en cursiva, las instrucciones correspondientes del lenguaje de programación:

- El usuario U1 envía un mensaje $M(U1>WS)$ al servidor WS:

15 $M(U1>WS) = \text{«Comand :move :user1 :robot3 :0.1,0.0,0.3»}$

- Se efectúa entonces un tratamiento $T(WS)$ dentro del servidor WS, para constituir C el mensaje según la estructura objeto predefinida:

Type = commande

Sous-type = move

Target = robot3

Source = user1

Size = 11

20 *Data = "0.1,0.0,0.3"*

- Seguido de una etapa denominada de serialización SE y de firma SI del mensaje:

087aa394104a8b6d9a3c06d35c768a0bcefadc9

{ "py/object": "message.message ",

"src": "web-user1",

"protocol": 0,

"target": { "py/tuple": [10.12.1.131, 48000] },

"type_msg": "command",

"sous_type": "move",

"size": 11,

25 *"data": "0.1,0.0,0.3" }*

- Se envía un mensaje $M(WS>R3)$ desde el servidor WS al robot R3.

- 30
- Se efectúa entonces un tratamiento $T(R3)_{P2}$ dentro del segundo programa del robot R3, para verificar VSI la firma y reconstituir REC el mensaje, que es transmitido a continuación por el segundo programa P2 del robot R3 al tercer programa P3 del robot R3, según el escenario previsto, es decir:

- según el modo "directo" ($M(R3;D)_{P2>P3}$) en el que se proporciona la orden/solicitud en forma adecuada para el robot (en el caso de un robot que permite la ejecución de órdenes por un sistema externo), directamente desde el segundo programa al tercer programa, por medio del archivo de configuración:

35 por ejemplo, para un robot con OR ROS:

p = ros.Publisher()

p.publish("/cmd_vel", msg.data)

- 40
- según el modo "interpretar" ($M(R3;I)_{P2>P4>P3}$) en el que se proporciona la información (de manera general) o la orden/solicitud (en el caso de robots que no permiten la ejecución de órdenes por un sistema externo) al robot desde el segundo programa al tercer programa a través del cuarto programa.

En este modo, el segundo programa envía el mensaje campo por campo (según la estructura objeto) al cuarto programa y el cuarto programa proporciona los elementos adecuados al tercer programa, generándose estos elementos adecuados dentro del cuarto programa y dependiendo de la estructura (esqueleto, programación prevista para los comportamientos a adoptar, etc.) de dicho cuarto programa.

Seguridad de los intercambios

La invención implica diversos canales de comunicación. Se enumeran, entre otros:

- comunicaciones según un protocolo de tipo UDP, relativas al primer programa P1 y/o al segundo programa P2,
- comunicaciones según un protocolo de tipo TCP, entre robots,
- comunicaciones según un protocolo de tipo UDP, entre la nube y los primer y segundo programas P1 y P2.

Ventajosamente, los diferentes canales de comunicación son seguros.

Son posibles diferentes elecciones técnicas, y el experto en la materia sabrá elegir las tecnologías más adecuadas para la aplicación considerada.

A título de ejemplo, se cita:

- en el caso de comunicaciones según un protocolo UDP:
 - el uso de enlaces directos ethernet,
 - el uso del protocolo DTLS ("Datagram Transport Layer Security" en inglés),
 - el uso de un conjunto de cifrado ("cipher suite" en inglés) que realiza, entre otros, unos algoritmos de cifrado tales como, por ejemplo, AES ("Advanced Encryption Standard" en inglés) y de hash tales como, por ejemplo, SHA ("Secure Hash Algorithm" en inglés)
- en el caso de comunicaciones según un protocolo TCP:
 - el cifrado y la autenticación por pares a través del protocolo "SSL" (capa de sockets protegidos), o
 - el cifrado y la autenticación por pares a través del protocolo "TLS" (seguridad de capa de transporte), a través de una realización libre (por ejemplo: OpenSSL) o no, y realizando un conjunto de cifrado ("cipher suite" en inglés), que permite la parametrización de la seguridad de dicho protocolo TLS a través de:
 - por un lado, un mecanismo de intercambio de claves, por ejemplo, mediante un cifrado asimétrico tal como RSA, o a través del algoritmo Diffie-Hellman, según un modo anonimizado o no,
 - por otro lado, unos mecanismos que aseguran la confidencialidad y la integridad de los datos intercambiados, por ejemplo a través de algoritmos de cifrado, tales como AES ("Advanced Encryption Standard" en inglés) y de hash tales como SHA ("Secure Hash Algorithm" en inglés)

Ejemplo de escenario de aplicación industrial

Estos procedimientos de interconexión y de gestión de la cooperación entre robots permiten, por ejemplo, la realización de un escenario tal como el presentado en las figuras 7 y 8.

La figura 7 presenta cuatro robots heterogéneos:

- R1: robot móvil con teledetección láser, con OS ROS
- R2: brazo manipulador con una pinza, con OS ABB
- R3: robot con brazo manipulador de marca diferente del robot R2 y OS ABB, y
- R4: un dron con OS Universal Robot

Cada robot utiliza su propio sistema operativo robótico.

Como se resume en el organigrama de la figura 8, en esta misión, el robot R1, provisto de un sensor de humedad, de un sensor de temperatura y de una etiqueta RFID, abandona su zona de partida Z1-ZD tras la recepción de un

mensaje por parte del robot R2 (provisto asimismo de una etiqueta RFID) que lo invita a reunirse en la zona Z2 que corresponde, por ejemplo, a una zona de carga ZCH. Para conseguirlo, el robot R1 debe efectuar una navegación autónoma utilizando, por ejemplo, los datos de su propia odometría y un telemetro láser. Una vez llegado a nivel de la zona Z2, el robot R1 debe comunicarse con un chip RFID que le indicará su zona de parada para la manipulación de las piezas.

5

En una segunda fase, notifica al robot R2 que está listo para recuperar los objetos a transportar. Así, el robot R2 recupera una pieza y la deposita sobre el robot R1, y después le informa del final de la carga.

10

El robot R1 se dirige después hacia la zona de tratamiento Z3-ZTR. Llegado al nivel de dicha zona Z3, notifica al robot R3 una información sobre la temperatura de la pieza.

Este último efectúa un "Tratamiento A" o un "Tratamiento B", según la temperatura de la pieza. Al final de esta fase, el robot R4 se incorpora a la zona Z3 y recupera la pieza para depositarla según su temperatura en la zona Z4A o Z4B. Mientras tanto, el robot R1 vuelve en la zona Z1 a la espera del próximo ciclo.

15

Así, como lo demuestra este ejemplo de escenario, los procedimientos según la invención permiten una comunicación fluida entre robots heterogéneos y sensores, proponiendo una arquitectura de software mayoritariamente genérica (capas P1, P2), fácil de instalación y de utilización (permaneciendo en su sitio el sistema operativo robótico inicial del robot) y un principio eficaz de conversión y de transmisión de mensajes que hace que la comunicación entre las capas de software superiores sea transparente.

20

REIVINDICACIONES

1. Procedimiento para la interconexión de un primer robot (1) con por lo menos un segundo robot (2, 21), comprendiendo cada robot (1, 2, 21) por lo menos:

- una unidad de control y mando, comprendiendo la unidad de control y mando por lo menos un procesador (4) y por lo menos un módulo de memoria (M);
- por lo menos un equipo periférico de tipo accionador o sensor;
- por lo menos un equipo de comunicación;
- un primer programa (P1), configurado para pilotar el por lo menos un equipo de comunicación, y que comprende una pluralidad de primeros módulos de software que se ejecutan en el por lo menos un procesador;
- un segundo programa (P2) que comprende una pluralidad de segundos módulos de software que se ejecutan en el por lo menos un procesador, siendo la configuración de dicho segundo programa (P2) determinada por un archivo de parametrización que comprende un campo typ_service que indica un modo de envío de entre los modos interpretar o directo, caracterizando dicho modo de envío los mensajes transmitidos por dicho segundo programa (P2).
- un tercer programa (P3) configurado para mandar por lo menos un accionador y/o por lo menos un sensor, y que comprende una pluralidad de terceros módulos de software que se ejecutan en el por lo menos un procesador;

comprendiendo el procedimiento las etapas siguientes, realizadas por el segundo programa (P2) dentro de cada uno de los robots (1, 2, 21), tras la recepción por parte del segundo programa (P2) de un primer mensaje procedente del tercer programa (P3):

- conversión del primer mensaje en un segundo mensaje, siendo el segundo mensaje formateado según una estructura objeto predefinida que comprende un campo typ_msg que indica un tipo de dicho mensaje de entre los tipos: orden, solicitud, o información,
- transmisión del segundo mensaje a por lo menos un programa de entre: primer programa (P1) perteneciente al mismo robot, segundo programa (P2) de otro robot;

comprendiendo el procedimiento de interconexión además las etapas siguientes, realizadas por el segundo programa (P2) dentro de cada uno de los robots (1, 2, 21), después de la recepción por parte del segundo programa (P2) de un tercer mensaje, formateado según dicha estructura objeto predefinida, procedente de por lo menos un programa de entre: primer programa (P1) perteneciente al mismo robot, segundo programa (P2) de otro robot, cuando el campo typ_msg de dicho tercer mensaje indica el tipo orden o el tipo solicitud:

- traducción del tercer mensaje en una parte de programa apta para ser realizada por el tercer programa (P3) cuando el campo typ_service del archivo de parametrización asociado a dicho tercer mensaje indica el modo de envío directo; y/o
- transmisión según un segundo protocolo de transporte, de por lo menos una parte de dicho tercer mensaje, o de dicha parte de programa, con destino al tercer programa (P3).

2. Procedimiento según la reivindicación 1, en el que la etapa de traducción comprende el acceso dentro del segundo programa a un archivo de configuración registrado en por lo menos un módulo de memoria (M), proporcionando dicho archivo de configuración al segundo programa la parte de programa apta para ser realizada por el tercer programa (P3).

3. Procedimiento según una de las reivindicaciones 1 o 2, en el que la etapa de transmisión de por lo menos una parte de dicho tercer mensaje, o de dicha parte de programa, comprende por lo menos:

- una primera transmisión, según un segundo protocolo de transporte, de un cuarto mensaje del segundo programa (P2) hacia un cuarto programa (P4),
- una segunda transmisión de un quinto mensaje de dicho cuarto programa (P4) con destino al tercer programa (P3).

4. Procedimiento según una de las reivindicaciones anteriores, en el que se utiliza un primer protocolo de transporte para la transmisión de un sexto mensaje entre el segundo programa P2 del primer robot (1, 2, 21) y el

segundo programa P2 del segundo robot (1, 2, 21).

5. Procedimiento según una de las reivindicaciones anteriores, en el que el primer, segundo, tercer, cuarto, quinto y sexto mensajes intercambiados están en formato JSON.

6. Procedimiento según una de las reivindicaciones 3 a 5, en el que el primer protocolo es del tipo UDP.

7. Procedimiento según una de las reivindicaciones 1 a 6, en el que el segundo protocolo es del tipo TCP.

8. Procedimiento según una de las reivindicaciones anteriores, en el que el por lo menos un equipo de telecomunicaciones comprende un medio de comunicación inalámbrica.

9. Procedimiento según una de las reivindicaciones anteriores, en el que el medio de comunicación inalámbrica comprende por lo menos un módulo de entre: módulo de tipo WIFI, módulo de tipo RFID, módulo de tipo Bluetooth, módulo de tipo GSM.

10. Procedimiento según una de las reivindicaciones anteriores, en el que el segundo programa (P2) cifra el segundo mensaje antes de transmitirlo si el destinatario es el por lo menos un segundo robot (2), y en el que el segundo programa (P2) descifra el tercer mensaje antes de traducirlo o de transmitirlo, si el expedidor es el por lo menos un segundo robot (2).

11. Procedimiento según una de las reivindicaciones anteriores, en el que el por lo menos un segundo robot comprende un operario remoto conectado a un servidor Internet.

12. Procedimiento según una de las reivindicaciones anteriores, en el que el tercer programa (P3) comprende el sistema operativo del robot y un programa piloto de por lo menos un accionador (A1, A2) y/o de por lo menos un sensor (C).

13. Procedimiento para la gestión de la cooperación entre una pluralidad de robots (1, 2, 12), comprendiendo cada robot de la pluralidad de robots (1, 2, 21) por lo menos:

- una unidad de control y mando, comprendiendo la unidad de control y mando por lo menos un procesador (4) y por lo menos un módulo de memoria (M);
- por lo menos un equipo periférico de tipo accionador (A1, A2) o sensor (C);
- por lo menos un equipo de comunicación conectado a por lo menos una red de comunicaciones;
- un primer programa (P1), configurado para pilotar el por lo menos un equipo de comunicación, y que comprende una pluralidad de primeros módulos de software que se ejecutan en el por lo menos un procesador;
- un segundo programa (P2) que comprende una pluralidad de segundos módulos de software que se ejecutan en el por lo menos un procesador,
- un tercer programa (P3) configurado para mandar el por lo menos un accionador y/o el por lo menos un sensor, y que comprende una pluralidad de terceros módulos de software que se ejecutan en el por lo menos un procesador;

comprendiendo el procedimiento las etapas siguientes:

- emisión, por una capa aplicativa alojada por un servidor (3) o accesible por Internet, de un archivo de misión, comprendiendo dicho archivo de misión una lista de tareas elementales planificadas y su secuenciación, y de un archivo de identificación y de priorización de los equipos periféricos,
- recepción de dicho archivo de misión y de dicho archivo de identificación y de priorización por un servicio equipo del primer programa (P1) de cada robot de la pluralidad de robots,
- transmisión, por el primer programa (P1) de cada robot de la pluralidad de robots, de un primer mensaje de tipo solicitud con destino a su segundo programa (P2), para conocer un identificador de cada periférico (A1, A2, C) disponible del por lo menos un periférico de cada robot de la pluralidad de robots (1, 2, 21),
- realización, por el segundo programa (P2) de cada robot de la pluralidad de robots (1, 2, 21), del procedimiento para la interconexión de robots según una de las reivindicaciones anteriores para solicitar al tercer programa (P3) y transmitir al primer programa (P1) los identificadores de los periféricos (A1, A2, C) disponibles en cada robot de la pluralidad de robots (1, 2, 21);

- 5
 - tratamiento, por el servicio de equipo del primer programa (P1) de cada robot de la pluralidad de robots (1, 2, 21), de los identificadores de los periféricos (A1, A2, C) disponibles para generar una lista de las tareas realizables por cada robot (1, 2, 21) de la pluralidad de robots (1, 2, 21),
 - transmisión a cada robot (1, 2, 21) de la pluralidad de robots (1, 2, 21) y/o a un servicio centralizado del servidor, por cada servicio negociación/elección del primer programa P1, de dicha lista de las tareas realizables por cada robot de la pluralidad de robots (1, 2, 21),
- 10
 - producción, por un servicio centralizado del servidor y/o tras un proceso de comunicación entre cada servicio de negociación/elección de los robots (1, 2, 21), en base al archivo de identificación y de priorización de los periféricos y de la lista de las tareas realizables, de una lista de los robots que participan en la misión,
- 15
 - ejecución de la misión,
 - sincronización de los robots,
 - supervisión de la ejecución de la misión.
- 20

14. Programa de ordenador que comprende un conjunto de instrucciones configuradas para ejecutar el procedimiento para la interconexión de robots según una de las reivindicaciones 1 a 12.
- 25

15. Programa de ordenador que comprende un conjunto de instrucciones configuradas para ejecutar el procedimiento de gestión de cooperación según la reivindicación 13.

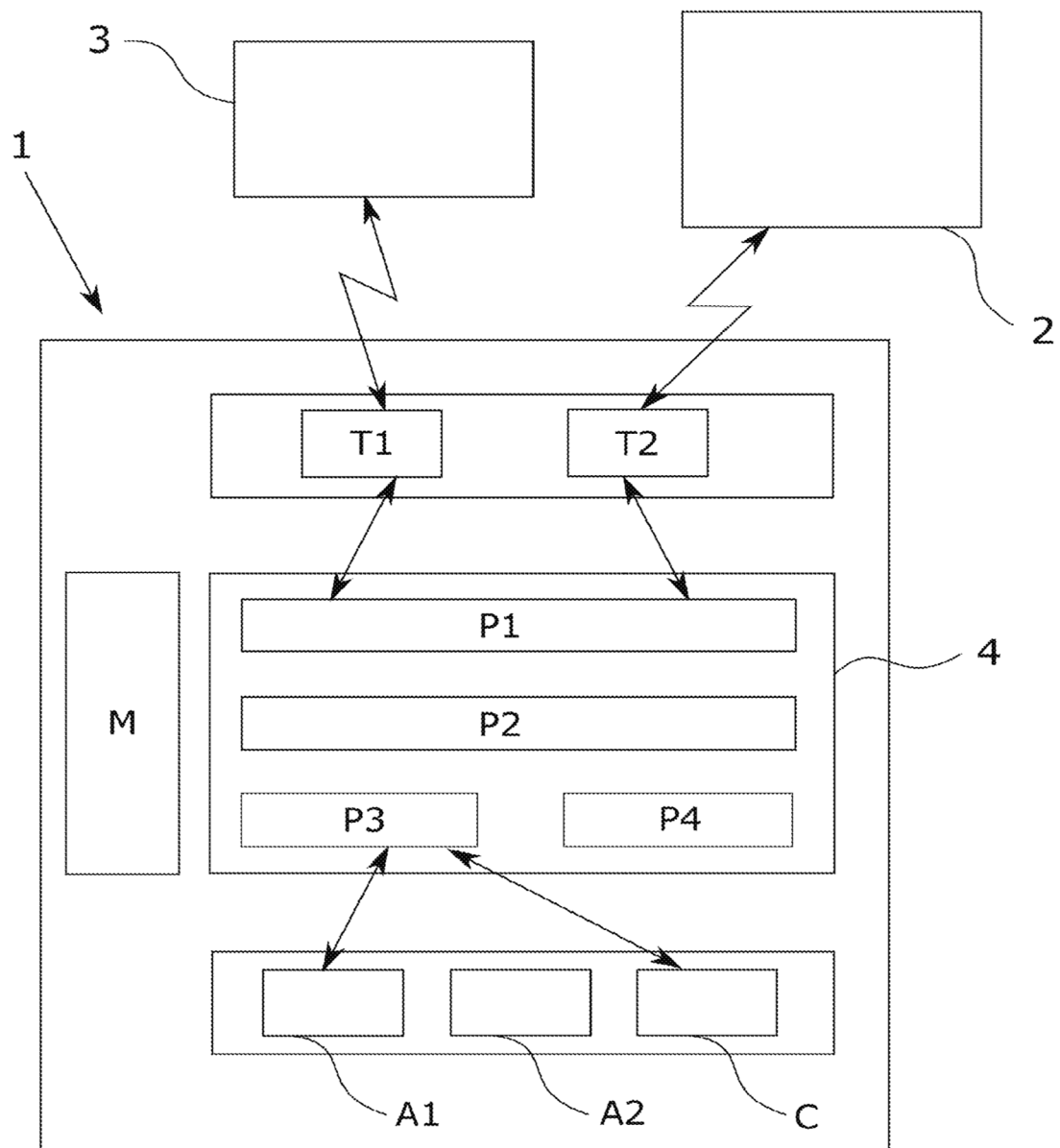


Fig.1

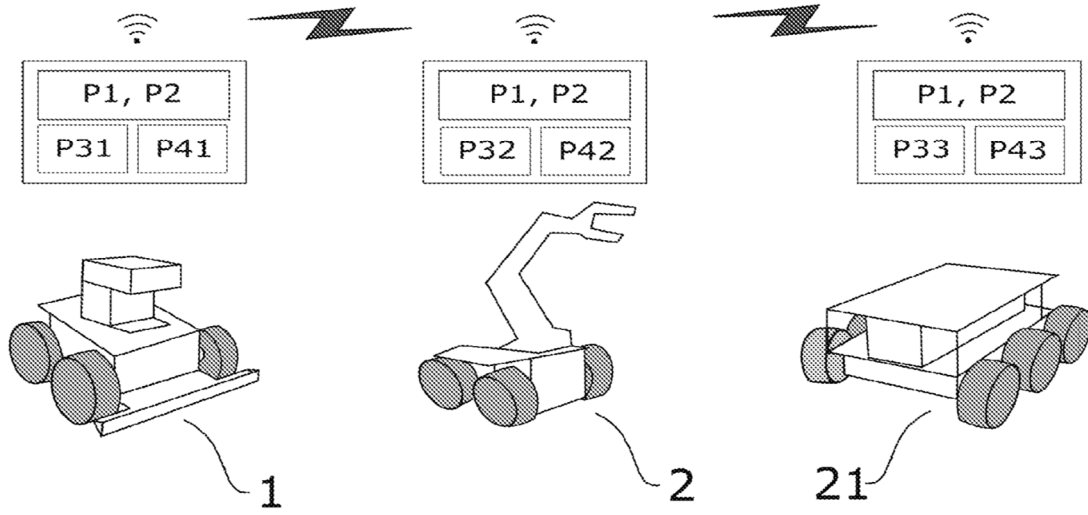
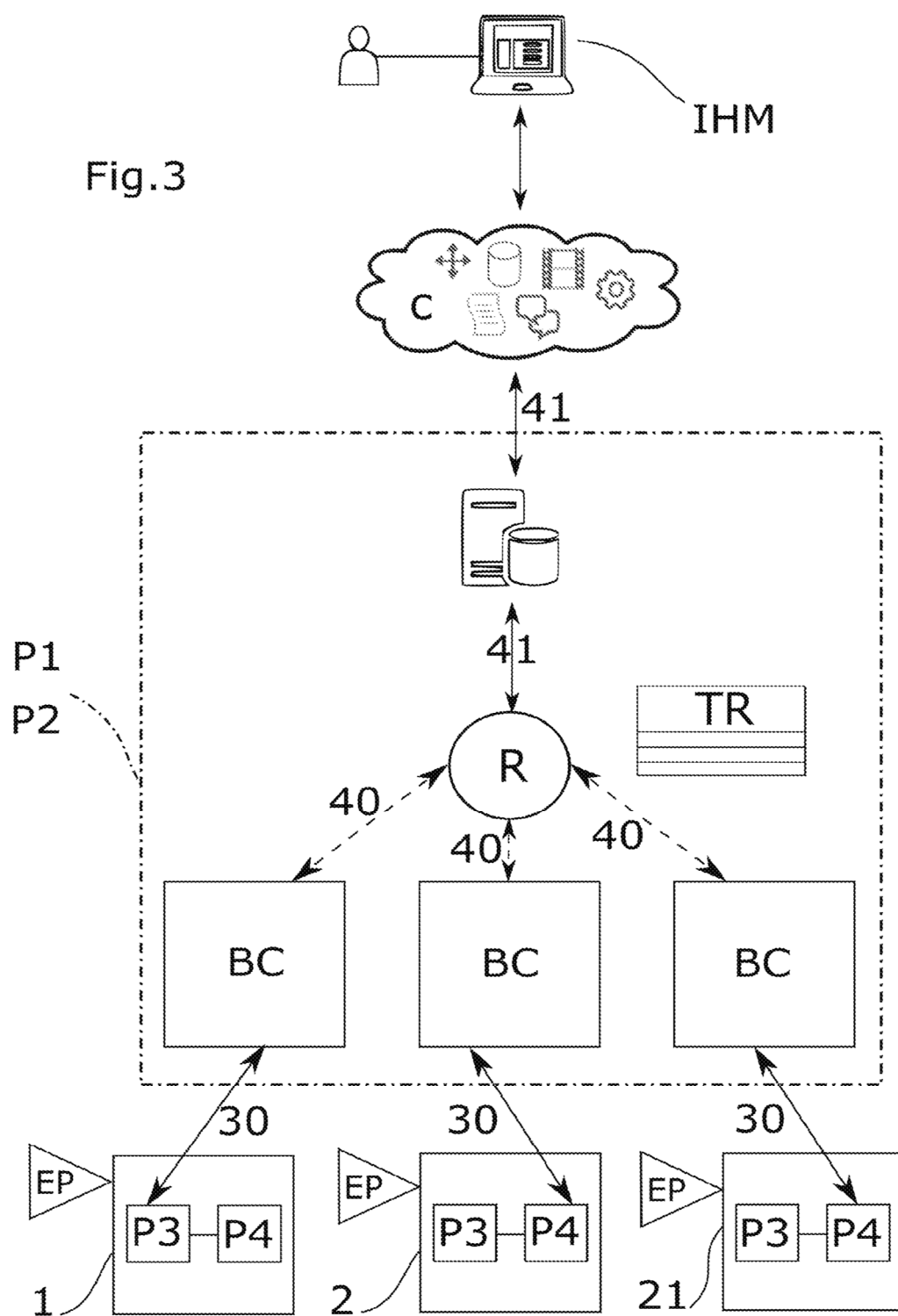


Fig.2



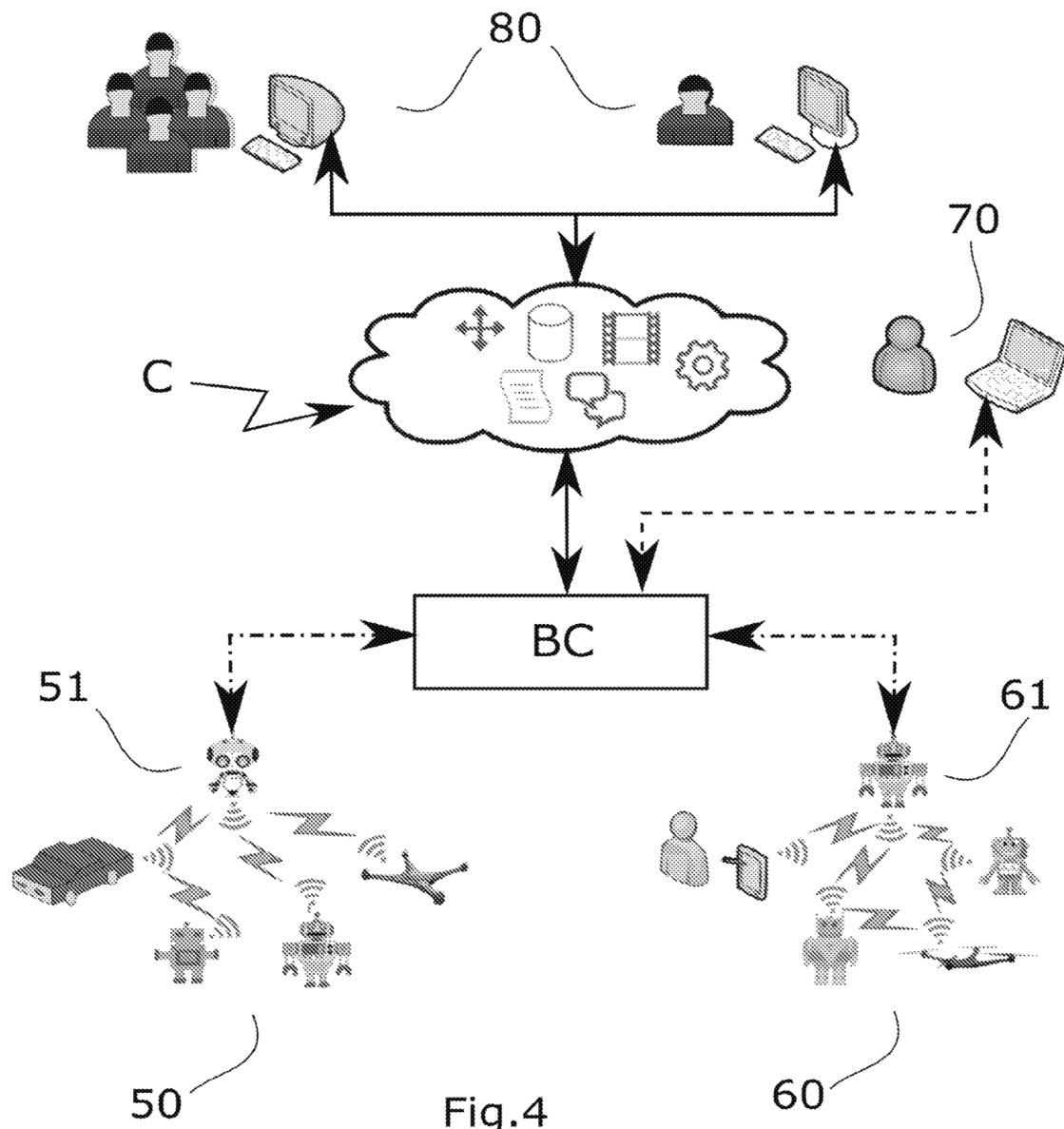


Fig.4

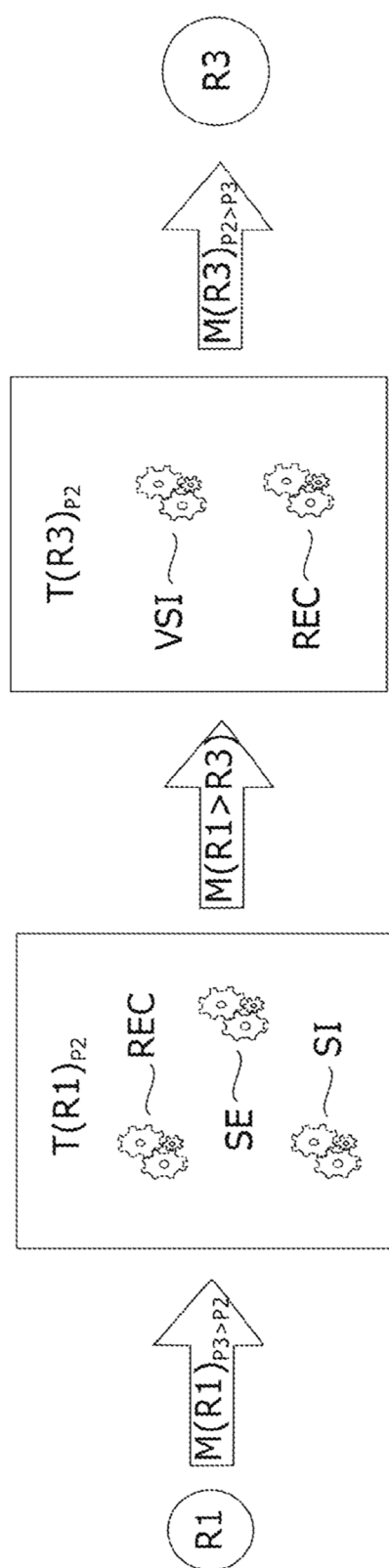


Fig.5

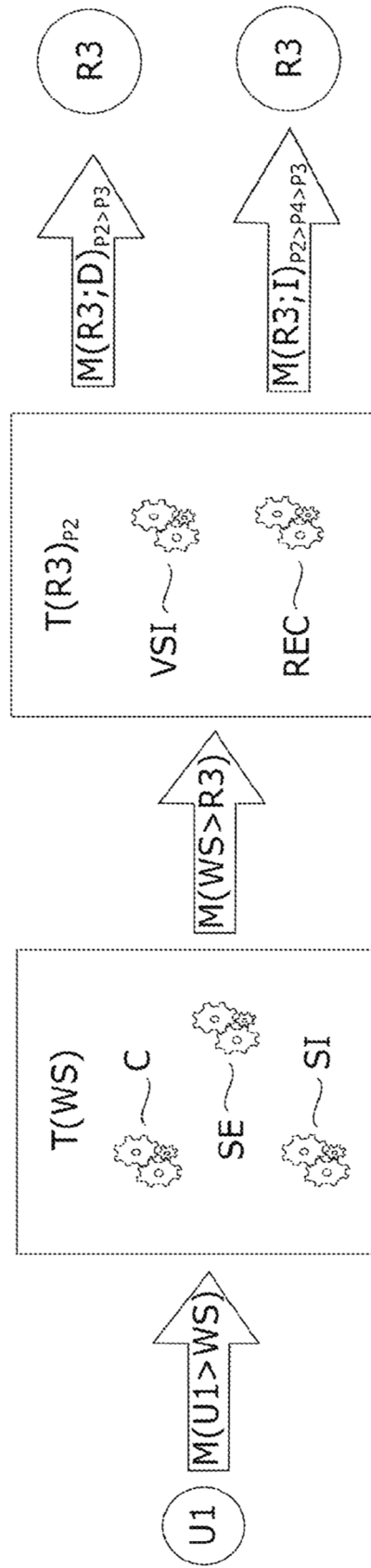


Fig.6

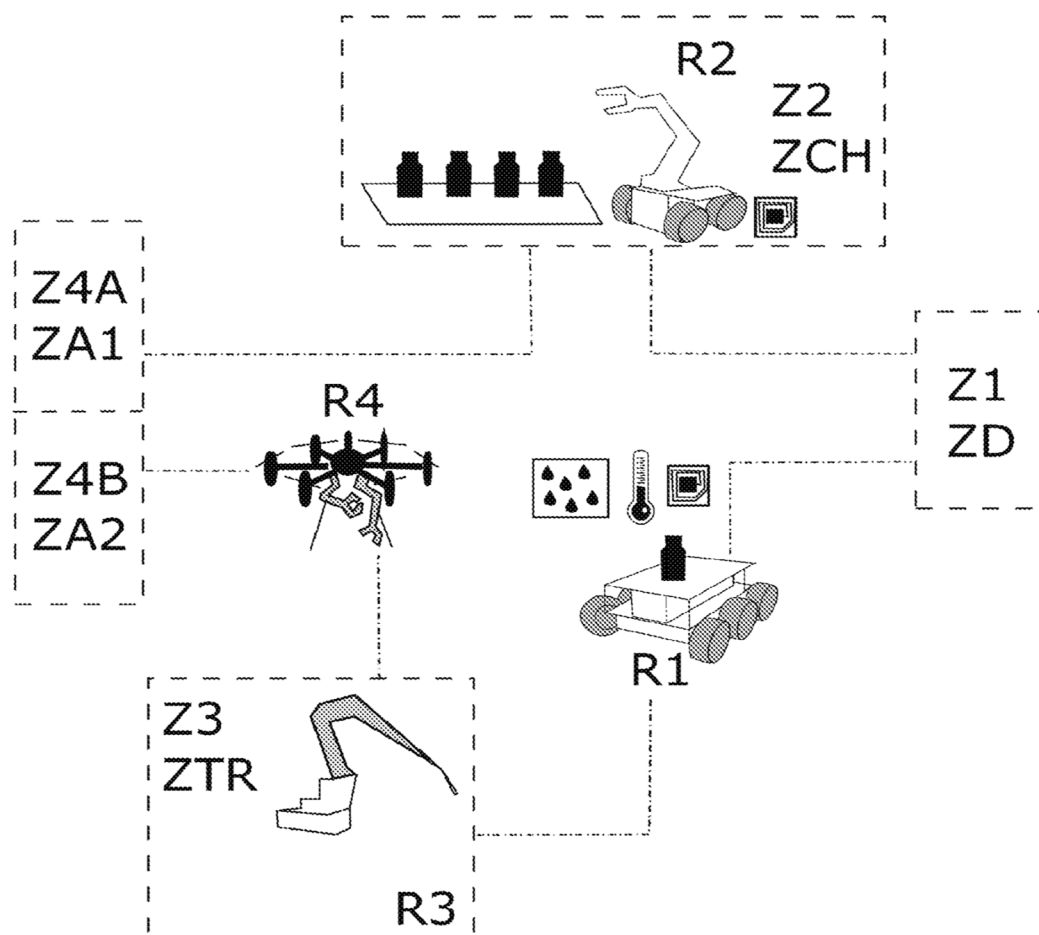


Fig.7

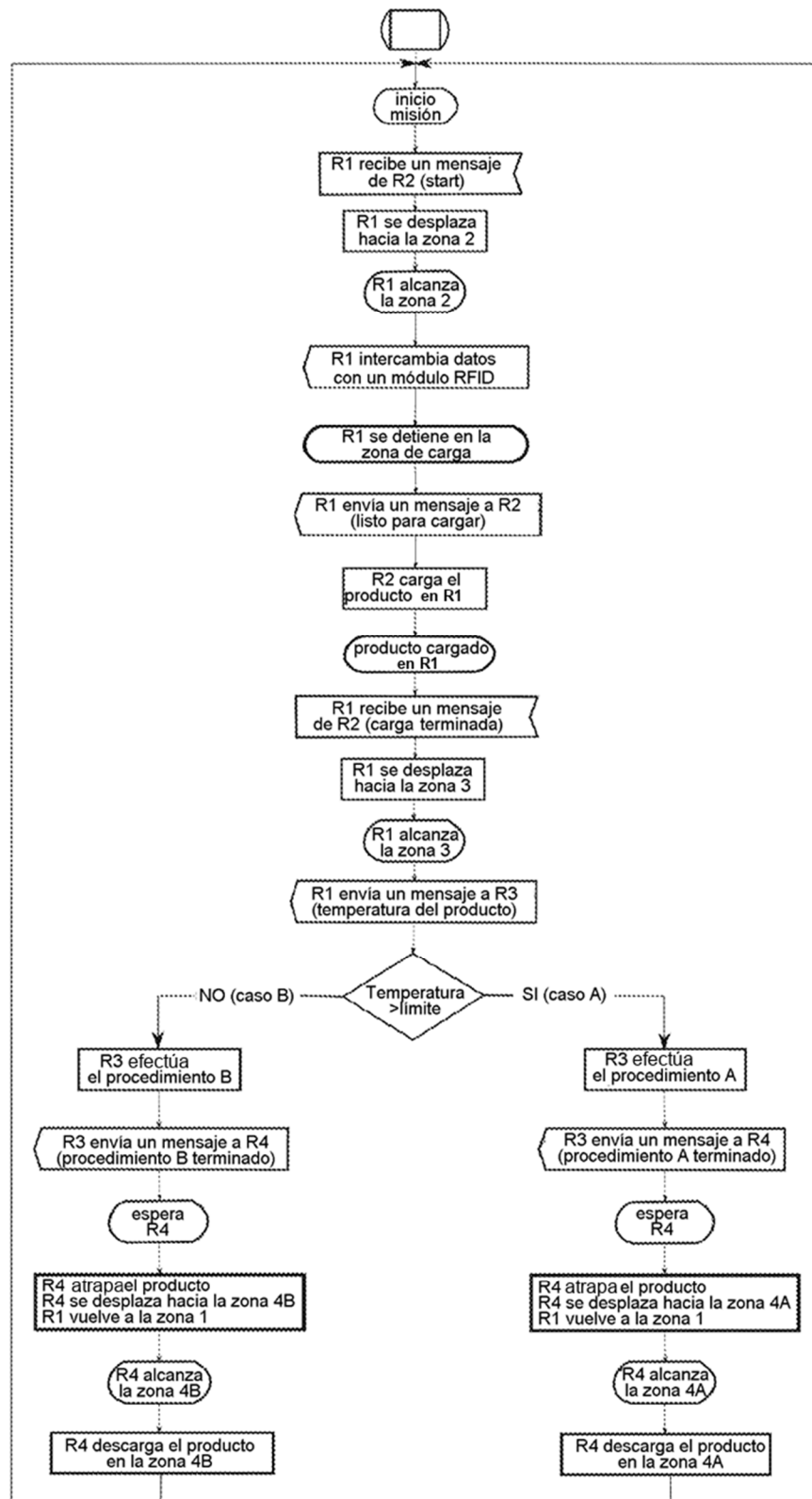


Fig.8