



Universidad de Castilla-La Mancha

Escuela Superior de Informática

Departamento de Tecnologías
y Sistemas de Información

Programa Oficial de Postgrado en Tecnologías Informáticas Avanzadas

Trabajo Fin de Máster

Framework de Desarrollo para Sistemas Multi-Robot

Septiembre de 2010

Alumno: José Carlos Hidalgo Nevado
Directores: Dr. David Vallejo Fernández
D. Eduardo Domínguez Parra

Resumen

En la actualidad, el interés por los sistemas multi-robot ha crecido frente al desarrollo de un único robot debido a las ventajas que ofrece a la hora de abordar problemas complejos en entornos dinámicos y heterogéneos. El desarrollo de este tipo de sistemas supone un gran desafío para la comunidad robótica ya que supone un transición de los sistemas centralizados a los sistemas distribuidos.

El desarrollo de estos sistemas robóticos no es una tarea trivial por multiples aspectos. En este contexto, destacan el acceso a los dispositivos físicos, la incorporación de nuevos dispositivos, la tolerancia a fallos, la latencia en las comunicaciones y la gestión y coordinación de robots heterogéneos. Además, el incremento de la complejidad de los comportamientos inteligentes supone que el desarrollo de este tipo de sistemas necesite de un soporte que facilite su diseño e implementación.

Este trabajo propone un framework para el desarrollo de sistemas multi-robot que permite a los desarrolladores disponer de la infraestructura de comunicaciones necesaria para desarrollar estos tipos de sistemas más fácilmente, centrando el trabajo en la capa de aplicación y delegando las comunicaciones en el propio framework. El uso de un middleware de comunicaciones orientado a objetos complementa al framework con servicios relevantes tales como la localización transparente o la publicación/subscripción de eventos. Por último, se describen dos casos de estudio que permiten mostrar el uso de este framework para abordar el desarrollo de distintos sistemas robóticos.

Índice

Resumen	I
1. Introducción	1
1.1. Motivación	2
1.2. Objetivos	2
1.3. Estructura del documento	3
2. Estado del arte	4
2.1. Frameworks de comunicaciones para sistemas reactivos	4
2.2. Arquitecturas específicas para sistemas multi-robot	6
2.2.1. Recogida de objetos	6
2.2.2. Observación de múltiples objetivos	7
2.2.3. Movimiento de objetos	8
2.2.4. Exploración	9
2.2.5. Fútbol robótico	9
2.2.6. Discusión	11
3. Descripción de la Arquitectura Propuesta	12
3.1. Capa SIAN	12
3.2. Capa SBN	15
3.3. Casos de Estudio	16
3.3.1. Control de un robot móvil autónomo	17
3.3.2. Coordinación de un sistema multi-robot	19

4. Conclusiones y Líneas de Investigación Futuras	20
4.1. Conclusiones	20
4.2. Líneas futuras de investigación	21
Bibliografía	23
A. Resumen de las asignaturas cursadas en el Máster	27
A.1. Redes Multimedia	27
A.2. Sistemas distribuidos avanzados: grid e inteligencia ambiental	28
A.3. Sistemas heterogéneos en red	29
A.4. Pruebas y seguridad de Sistemas de Información	29
A.5. Cognición y colaboración	30
A.6. Técnicas de softcomputing	31
B. Curriculum Vitae	32

Índice de figuras

3.1. Visión general de la arquitectura propuesta.	13
3.2. Robot Móvil Autónomo ATOS I.	18
3.3. Esquema de conexiones de los dispositivos en la capa SBN.	18
3.4. Utilización del framework propuesto en una arquitectura multi-agente.	19

Capítulo 1

Introducción

En estos últimos años, el desarrollo de sistemas multi-robot (en adelante SMR) ha cobrado un especial interés dentro del campo de la robótica. Dicho desarrollo es consecuencia, entre otras muchas razones, de las capacidades que estos sistemas poseen frente al desarrollo de un único robot (robots más complejos) . Entre estas capacidades destacan principalmente el comportamiento cooperativo, la tolerancia a fallos, la paralelización de tareas y la escalabilidad [1] .

Un sistema multi-robot se puede definir como un conjunto de robots (homogéneo o heterogéneo), donde cada elemento del sistema tiene definido un plan para alcanzar unos objetivos o metas globales [36]. Esta división en subtarefas permite realizar una paralelización acortando así el tiempo necesario para alcanzar el objetivo. Entre las tareas desarrolladas por los sistemas multi-robot destacan la exploración de entornos parcial o completamente desconocidos, la vigilancia, la búsqueda y el rescate, la manipulación distribuida (teleoperación), la fusión sensorial y la localización distribuida, el transporte de objetos y las competiciones de *Robocup*[®] ¹(competiciones de fútbol robótico).

La mayoría de los sistemas multi-robot pueden entenderse como sistemas reactivos complejos [10][9], ya que deben responder a acontecimientos externos en función de su estado. Esta reactividad, se suele complementar con un componente deliberativo, que proporciona al sistema robótico un comportamiento inteligente.

En una buena parte de los casos este comportamiento deliberativo lo proporciona una capa formada por agentes software [37]. Algunos autores [39] relacionan el concepto de coordinación de los SMR con la coordinación en los sistemas multi-agente (SMA), ya que éstos también tienen la capacidad de coordinar recursos heterogéneos para conseguir un fin común. Por ello, la mayoría de SMR utilizan agentes software como soporte en la toma de decisiones o como componentes del propio sistema [28].

¹<http://www.robocup.org/>

1.1. Motivación

El modelado de esta clase de sistemas robóticos implica generalmente cuestiones de diseño e implementación de capas cercanas al hardware, que son responsables de la integración de los dispositivos de entrada/salida (sensores y actuadores).

En estos últimos años, diferentes componentes hardware se han utilizado para integrar dispositivos de entrada/salida en los sistemas robóticos [36]. Sin embargo, el uso de estos componentes hardware conlleva múltiples limitaciones. En primer lugar, existe una dependencia entre el sistema operativo y el driver que gestiona el dispositivo de adquisición de datos. El concepto de escalabilidad también puede convertirse en un problema ya que las interfaces de comunicaciones y los dispositivos compatibles están generalmente muy limitados.

Si la integración de nuevos dispositivos plantea numerosos problemas, el acceso a la información procedente de los dispositivos también puede representar un desafío. Cuando los procesos o agentes acceden de forma local a los dispositivos (por ejemplo dentro de la misma plataforma hardware), para resolver este acceso se suelen usar las llamadas al sistema o bibliotecas adecuadas para obtener información de los sensores o para actuar sobre los actuadores. El problema surge cuando los dispositivos están distribuidos y, por lo tanto, este acceso es remoto; este problema es provocado por la falta de estándares y la dependencia inducida por los sistemas operativos utilizados en los sistemas desplegados.

Por lo tanto, el control y la coordinación de equipos de robots y/o agentes software no es una tarea trivial, ya que en muchos casos, estos sistemas pueden estar formados por robots o recursos heterogéneos. En este contexto, el desarrollo de un framework para modelar sistemas multi-robot puede contribuir a centrar el trabajo del investigador en el dominio de aplicación simplificando el tratamiento de los problemas de comunicación.

1.2. Objetivos

Teniendo en cuenta los problemas anteriores, en este Trabajo Fin de Máster se describe una solución integral hardware/software que permite el acceso remoto a los dispositivos, con gran independencia de sistemas operativos o lenguajes de programación, y que además permite la incorporación de nuevos dispositivos, como robots o simples sensores/actuadores, simplificando así el desarrollo de sistemas más complejos.

La solución propuesta se basa en el framework ATOS [5], una arquitectura genérica que permite la integración y el acceso remoto a dispositivos de entrada/salida. Tomando como base esta arquitectura, se realizarán las modificaciones necesarias

para adaptar dicho framework a las necesidades particulares de un SMR. Con el uso de este framework, los desarrolladores pueden definir y utilizar interfaces, dentro de un entorno distribuido, que identifican robots o sensores/actuadores desde un punto de vista abstracto, sin necesidad de preocuparse por los cambios que se puedan realizar o cómo se incorporan aquellos al sistema.

1.3. Estructura del documento

El resto del trabajo está estructurado de acuerdo con los siguientes puntos:

- **Capítulo 2.** En esta sección se realiza un estudio de las distintas arquitecturas específicas para sistemas multi-robot dependiendo de los diferentes ámbitos de aplicación de estos sistemas y centrando dicho estudio en los mecanismos de comunicación utilizados por dichas arquitecturas. Además, se estudian distintas arquitecturas genéricas que ofrecen un soporte de comunicaciones para el desarrollo de sistemas reactivos.
- **Capítulo 3.** En esta sección se expone el trabajo de investigación realizado hasta el momento actual, donde se describen las distintas capas que componen la arquitectura propuesta y varios casos de estudio donde se detallan distintos ejemplos de implementación.
- **Capítulo 4.** En esta sección se comentan las conclusiones obtenidas a partir de la elaboración de este trabajo y se proponen las posibles líneas de investigación futuras que surgen a partir del mismo.
- **Bibliografía:** En esta sección se anotarán las referencias bibliográficas utilizadas a lo largo del texto.
- **Anexo A:** Resumen de las asignaturas cursadas en el Máster.
- **Anexo B:** Curriculum Vitae.

Capítulo 2

Estado del arte

En este capítulo se realiza una revisión del estado del conocimiento de las diferentes arquitecturas para sistemas multi-robot. En un primer apartado, se realizará un estudio de los frameworks de comunicaciones utilizados en sistemas robóticos o similares. A continuación, se estudiarán las arquitecturas específicas para el desarrollo de sistemas multi-robot dependiendo del ámbito de aplicación de los mismos, analizando los distintos mecanismos de comunicación utilizados en ellas.

2.1. Frameworks de comunicaciones para sistemas reactivos

El soporte genérico de comunicaciones utilizado en las arquitecturas de sistemas reactivos, como por ejemplo, los sistemas domóticos, redes de sensores (WSN), etc., son muy similares a los utilizados en los SMR.

En este sentido, la arquitectura para el control distribuido de un sistema robótico desarrollada por Pertersson et al. [27] usa el middleware [13] para distribuir las tareas de control de un sistema robótico. Esta arquitectura permite realizar una paralelización de las tareas de control y que éstas sean accedidas a través de módulos de alto-nivel.

Magenat et al. [18] proponen una arquitectura modular diseñada para comunicar múltiples dispositivos de entrada/salida. Esta arquitectura permite que procesos externos accedan a la información ofrecida por los dispositivos físicos utilizando conexiones TCP/IP.

Otra propuesta muy interesante es el middleware para robots móviles (MIRO) propuesto por Utz et al. [33]. Dicho middleware permite obtener información de

los sensores o interactuar con los actuadores de una plataforma robótica utilizando el middleware CORBA.

Orebäck et al. [24] [23] estudiaron diferentes soluciones arquitectónicas para el desarrollo de sistemas robóticos, proponiendo finalmente el framework ORCA, que fue desarrollado bajo el enfoque de la ingeniería de software basada en componentes [32]. En estos trabajos, los autores introducen el concepto de *contenedores de hardware* para referirse al acceso básico a los dispositivos físicos.

En el campo de los sistemas multi-agente [37], Posadas et al. [28] han sugerido, a través de la estructuración de sistemas robóticos, una arquitectura híbrida basada en agentes, con múltiples buses de comunicación para interactuar con las capas reactiva y deliberativa. En este sentido, Yang et al. [40] proponen otra arquitectura híbrida donde los sensores y actuadores se hacen corresponder con agentes que se comunican entre sí.

Quigley et al. [29] van un paso más allá y proponen un sistema operativo *Open Source* para robots, denominado ROS, que puede clasificarse como un framework de desarrollo que ofrece las herramientas y bibliotecas necesarias para el desarrollo de sistemas robóticos. ROS permite realizar una abstracción del hardware y una reutilización e integración de robots y dispositivos, encapsulando éstos tras interfaces estables. Además, proporciona una capa de comunicaciones entre componentes o agentes del sistema robótico y mecanismos de paso de mensajes *RPC* o *Publish/Subscribe*, *Service lookup*, etc. Otras propuestas similares son los frameworks Player/Stage [6], Orocos¹ o Carmen [19].

Por último, Moya et al. [21] describen cómo integrar un middleware orientado a objetos dentro de dispositivos con recursos limitados (denominados *picoObjetos*) utilizados en las redes inalámbricas de sensores (WSN). En concreto, los autores han realizado varias implementaciones de middlewares como CORBA (denominado picoCORBA) o ICE [11] (denominada picoICE).

En general, la mayoría de estos frameworks proponen la utilización de una capa de alto nivel para abstraer al cliente de la comunicación con los dispositivos a través del uso de interfaces. Sin embargo, muchos de ellos [27][33] no tienen en cuenta la comunicación entre los componentes de más bajo nivel frecuentemente utilizada en los SMR. Además, en algunos casos [40] los dispositivos de bajo nivel están conectados a la capa de alto nivel a través de tarjetas de adquisición o similares que permiten un acceso directo a los dispositivos pero que por el contrario, tienen una gran dependencia del sistema operativo donde se ejecutan y del número de interfaces disponibles para la conexión de dispositivos físicos. Otras arquitecturas [21] que evitan el uso de estos dispositivos de comunicación empleando microncontroladores como servidores hardware a través de conexiones de red, no ofrecen, por

¹The Orocos Project. <http://www.orocos.org>

el contrario, el soporte necesario para que agentes remotos puedan comunicarse con los dispositivos.

2.2. Arquitecturas específicas para sistemas multi-robot

Dependiendo del ámbito de aplicación del sistema multi-robot, se pueden identificar diferentes tipos de arquitecturas, ya que las diferentes tareas realizadas en cada ámbito determinan los recursos que necesita cada tipo de sistema. Los ámbitos de aplicación más comunes [8] se describen a continuación:

2.2.1. Recogida de objetos

En este ámbito de aplicación se realizan tareas tales como operaciones de rescate y búsqueda, limpieza de residuos tóxicos, etc.. En este dominio se requieren robots inteligentes capaces de identificar y recoger objetos dispersos en un entorno parcial o completamente desconocido. En este tipo de aplicaciones, normalmente se utiliza un soporte externo que permite la identificación de objetos o el posicionamiento dentro de un entorno.

En 1998, Parker propuso una arquitectura software distribuida llamada ALLIANCE [25] que facilita la tolerancia a fallos en la cooperación de un conjunto de robots heterogéneos. En este caso, se trata de una arquitectura muy genérica que puede utilizarse para implementar distintos comportamientos. Su implementación consiste en el desarrollo de un sistema multi-robot destinado a la recogida de objetos peligrosos. ALLIANCE permite que los robots del sistema respondan de manera robusta, flexible y coherente a los cambios y modificaciones del entorno o del equipo de robots (fallos mecánicos, incorporación de nuevos robots, eliminación de componentes, etc.). La comunicación entre los distintos robots depende de las interfaces utilizadas, teniendo que desarrollar los módulos correspondientes para que los robots se puedan comunicar en cada caso. Esta arquitectura sólo soporta un envío de mensajes *one-way* entre los distintos robots.

En [2] se describe una arquitectura multi-agente para dar soporte a la tarea de recogida de objetos mediante un sistema multi-robot. Esta arquitectura permite la búsqueda, recogida y entrega en un punto de destino de múltiples objetos, evitando los posibles obstáculos que pudieran existir en el entorno. En esta arquitectura, los agentes software se encargan de realizar las tareas más complejas como la identificación del tipo de objeto y el posicionamiento en el entorno. Por el contrario, los robots ejecutan las ordenes procedentes de los agentes como la recogida de un

objeto o la evasión de obstáculo. Para la implementación se ha utilizado la herramienta JavaBots [3] que permite realizar una simulación del sistema y, además, ofrece una API para implementar las comunicaciones con los distintos robots que forman el sistema multi-robot. JavaBots está estructurado como un conjunto de clases Java y APIs que permiten desarrollar, ejecutar o simular sistemas de control en robots móviles.

Por último, Jung [16] describe una arquitectura basada en agentes con un comportamiento definible para la coordinación multi-robot; como ejemplo de implementación se utiliza la arquitectura para realizar tareas de limpieza de entornos. En este caso, se propone un esquema en capas para implementar las comunicaciones. Este esquema en capas permite diferenciar distintos casos o comportamientos de comunicación con los robots; el tipo de comunicación dependerá de la capa en la que se encuentren los agentes responsables. Por ejemplo, si dos robots necesitan comunicarse y no existe contacto visual entre ellos, pueden utilizar un sistema de mensajes por radiofrecuencia. En cambio, si los robots están cerca uno del otro pueden utilizar mecanismos de comunicación visual explícita entre ellos.

2.2.2. Observación de múltiples objetivos

La observación de múltiples destinatarios (también conocida como CMOMMT: Observación Cooperativa Multi-Robot de Múltiples Objetivos Móviles) es un ámbito de aplicación muy reciente en los SRM. La tarea consiste en maximizar el tiempo durante el cual cada uno de los objetivos en movimiento está siendo observado por, al menos, uno de los agentes robóticos del SMR. Hay muchas conexiones con la seguridad, la vigilancia y los problemas de reconocimiento. En este tipo de arquitecturas los problemas que se tratan de resolver están relacionados con la comunicación, la fusión de sensores, la cooperación y la coordinación.

Especialmente en el campo de la vigilancia, el paradigma multiagente está teniendo un especial interés. En este sentido, Vallejo et. al. [34] proponen una arquitectura multiagente para la vigilancia multi-robot en la cual un conjunto de robots obtiene el soporte necesario de un sistema multi-agente para realizar tareas de vigilancia y exploración de entornos. En este trabajo, el soporte proporcionado por la capa multiagente está orientado a la coordinación de los distintos robots. La estructura de esta arquitectura es de especial interés, ya que el modelo de comunicación está basado en el uso de canales de comunicación y de un esquema Publicador/Subscriptor, que garantiza la escalabilidad cuando aumente el número de robots. Además, permite asociar cada canal de comunicación a un determinado rol consiguiendo así un filtrado de los datos entre los distintos componentes del sistema. Otro aspecto muy importante, es el uso del servicio de publicación/subscripción entre los robots y los agentes software involucrados. Este servicio permite que los componentes suscritos a un determinado tipo de noticia obtengan

información cuando se publique algún cambio en algo que sea de su interés. Por ejemplo, cuando un robot detecta un objeto, éste publica este evento en el servicio de publicación y todos los demás componentes del sistema que estén suscritos a este tipo noticia tendrán conocimiento de dicho evento.

En el campo de la observación de múltiples destinos, Parker [26] propone una arquitectura muy similar a la arquitectura ALLIANCE, pero añadiendo el soporte necesario para realizar la búsqueda y seguimiento de un objeto. En el área de las comunicaciones no se realiza ninguna modificación, ya que todo el proceso de reconocimiento de objetos lo llevan a cabo los agentes software a través de un *streaming* de vídeo realizado por las distintas cámaras que se encuentra a bordo de los distintos robots.

Werger y Mataric [38] proponen una extensión bien definida de las técnicas de interacción de comportamientos PAB (*Port-Arbitrated Behavior*), que describe un estándar para intercambio de información a través de puertos mediante la definición de comportamientos sobre redes IP, que recibe el nombre de BLE (Broadcast of Local Eligibility). Esta técnica permite construir desde la abstracción sistemas multi-robot coordinados totalmente distribuidos.

2.2.3. Movimiento de objetos

En este ámbito, la tarea más común es el movimiento de cargas (normalmente cajas) dentro de un determinado entorno. Para este tipo de tareas los problemas más comunes se encuentran en la asignación de tareas, la robustez y en el área de las comunicaciones. Además, la coordinación es fundamental ya que inicialmente se necesita que los robots se posicionen adecuadamente frente el objeto que se desea mover y que, una vez posicionados, determinen la fuerza que deben aplicar al objeto para moverlo en la dirección deseada.

En el trabajo propuesto por Kube y Bonabeau [17] se describe una arquitectura basada en el transporte cooperativo de las hormigas. Este trabajo describe cómo un conjunto de robots se posicionan ante el objeto que desean mover a través de la interacción directa con los otros componentes del sistema. Después de haberse posicionado ante el objeto que se desea mover (mediante el envío de mensajes a través de las distintas interfaces de comunicación disponibles en los robots) se determina la dirección en la que se desea mover el objeto.

Otro ejemplo de arquitectura destinada al movimiento de objetos es la propuesta por Noreils [22]. En este trabajo se describe una arquitectura multiagente para la integración de la cooperación entre robots heterogéneos. El trabajo desarrolla un esquema basado en cuatro fases secuenciales, en las cuales se descompone la tarea a realizar, se realiza la asignación de tareas, su planificación y, por último, se

determina cuándo se deben de ejecutar estas tareas. Cada fase determina el tipo de comunicación entre los distintos componentes del SMR.

2.2.4. Exploración

Bajo el ámbito de la exploración se pueden agrupar diferentes tareas. Estas tareas difieren en la forma en que se realizan, pero comparten la característica común de que los miembros del SMR obtienen información del entorno. En la tarea de exploración existe la necesidad de distribuir a los robots por todo el entorno a monitorizar para evitar que se obtenga la misma información varias veces. La tarea de exploración puede ser de dos tipos, dependiendo de si el entorno es parcial o completamente desconocido. En las tareas de posicionamiento, siempre y cuando los robots no puedan obtener un posicionamiento absoluto por sí mismos, se utiliza un soporte externo que determinará cuál debe ser la posición de cada robot para evitar un solapamiento en la información obtenida.

Hugues realiza una implementación de un sistema multi-robot para realizar tareas de exploración [14], en la cual se describe un protocolo de comunicación para comunicar la información del entorno a agentes robóticos de alto-nivel. Los miembros del SMR cooperan en la construcción de un conocimiento común sobre el entorno que les rodea. Para ello, transmiten dicha información a una entidad de alto-nivel encargada de conservar únicamente la información más relevante dentro de la obtenida por el conjunto de robots. Todo este proceso se divide en cuatro sub-procesos: percepción del entorno, difusión de la información, filtrado y clasificación de la información obtenida.

Por último, Simmons et al. [31] proponen un conjunto de técnicas que permiten la coordinación multi-robot para realizar tareas de exploración y mapeado de un entorno. En este trabajo se describe un proceso que recibe el nombre de *ejecutivo* y que permite comunicar la capa de alto-nivel, formada por los componentes encargados de realizar la asignación y planificación de tareas, con una capa de exploración de bajo-nivel, formada por los robots encargados de realizar las tareas de exploración.

2.2.5. Fútbol robótico

En los últimos años, las competiciones de fútbol robótico (RoboCup) son, junto con la vigilancia, uno de los ámbitos más interesantes en el campo de la cooperación en sistemas multi-robot y multi-agente. De hecho, en los partidos de fútbol robótico la cooperación entre los robots es fundamental para un efectivo cumplimiento de la tarea en un entorno dinámico e incierto.

Dependiendo de las diferentes ligas de RoboCup, han sido estudiadas distintas cuestiones en la cooperación multi-agente. Por ejemplo, la liga de simulación² se refiere a la coordinación entre un grupo de agentes que ejecutan programas separados y que se comunican entre sí para alcanzar los objetivos marcados. Este tipo de escenario permite realizar las simulaciones necesarias sin necesidad de utilizar robots reales. Debido a la falta de información sobre el entorno, esta configuración proporciona un entorno interesante para los experimentos en la coordinación de un sistema multi-agente. Las otras ligas de RoboCup en cambio están compuestas por equipos de robots reales de diferentes tamaños en un entorno dinámico y desconocido a priori.

En la liga de robots de pequeño tamaño, los robots generalmente se controlan mediante un equipo remoto que puede utilizar un sistema de visión global (una cámara colocada sobre el campo). La disponibilidad de la información global del entorno y el uso de un equipo remoto para el control de los robots, permite la ejecución de las estrategias centralizadas [35]. En la mayoría de estos sistemas, el equipo remoto se puede considerar como un agente que actúa como el líder del equipo.

Por el contrario, en la liga de robots de tamaño medio, toda la sensorización debe ir a bordo de los robots. En este caso, la autonomía de los robots es mayor, ya que no dependen de elementos externos para obtener información del entorno. En este tipo de escenarios, normalmente, los equipos de robots son heterogéneos. Por lo tanto, es importante definir un protocolo de coordinación para que se puedan comunicar. En este sentido, Mota et al. [20] describen un esquema de comunicación para la coordinación de un equipo de fútbol robótico. Además, los autores proponen la definición de un vocabulario para describir las distintas tareas realizadas por un jugador, así como un conjunto de reglas que permiten gestionar la comunicación entre los jugadores. Yokota et al. [41] proponen un framework de comunicación para la coordinación de sistemas de robots autónomos distribuidos (DARS), donde se describe un protocolo basado en UDP para el intercambio de información entre los distintos jugadores. Además, describen otros tipos de comunicación más explícitos que permite que el SMR pueda hacer frente a los posibles fallos de comunicación. En este sentido, Roth et al. describen una propuesta de construcción en tiempo real de un modelo del entorno para un sistema multi-robot orientado a la competición *RoboCup*. En este trabajo los autores proponen dos tipos de modelos del entorno; uno individual, en el que se representa el estado de único robot, y otro compartido que representa el estado del conjunto de robots. Las comunicaciones entre ambos modelos son en tiempo real, consiguiéndose evitar el problema de la alta latencia en la comunicaciones mediante la compartición de información de los modelos. Otros ejemplos de arquitecturas de especial interés para la coordinación de equipos de fútbol robótico son [4] y [30].

²Introducción a la liga RobotCup. http://ciep.ing.uaslp.mx/jnunez/docs/robocup07_nunez.pdf

2.2.6. Discusión

En general las arquitecturas específicas para sistemas multi-robot proponen distintos soportes de comunicación entre los robots y agentes que son específicos para cada tipo de sistema o ámbito de aplicación [22][41]. En este sentido, todos los trabajos están orientados a la capa de aplicación pasando la comunicación con los robots a un segundo plano, ya que el tipo de comunicación dependerá de los robots utilizados en cada sistema multi-robot. El hecho de depender de los robots que se deseen utilizar en el SMR supone que para la implementación de una determinada arquitectura se tendrán que desarrollar los *drivers* necesarios en ambas capas para que la comunicación entre los robots y los agentes sea posible [31][14].

Por el contrario, cuando los SMR están formados por robots heterogéneos los problemas se multiplican, ya que al no existir una homogeneidad en los interfaces de comunicación y en los microcontroladores utilizados en los distintos robots, la mayoría de las arquitecturas sólo ofrecen soporte para una determinada interfaz de comunicación y para un determinado tipo de robot [25]. Otros problemas de la mayoría de estas arquitecturas son la latencia en las comunicaciones [2], la tolerancia a fallos [35] y la incorporación dinámica de nuevos robots al sistema [16].

Capítulo 3

Descripción de la Arquitectura Propuesta

El framework propuesto, como se ha comentado anteriormente, es una extensión del framework ATOS para el desarrollo de sistemas multi-robot. La arquitectura propuesta consiste en dos capas software distribuidas (SIAN y SBN) que permiten el intercambio de información entre los distintos elementos del SMR. La capa de Servicio de Bajo Nivel (SBN) está formada por los distintos componentes (normalmente robots) encargados de la interacción directa con los dispositivos físicos (sensores y actuadores). Por el contrario, la capa de Servicio de Alto Nivel (SIAN) se encarga de las cuestiones relacionadas con el acceso de clientes remotos (normalmente un soporte multiagente) a los distintos componentes de la capa SBN. Ambas capas están interconectadas a través de enlaces de red. La Figura 3.1 muestra la estructura de la arquitectura propuesta.

3.1. Capa SIAN

La capa SIAN puede entenderse como una fachada que permite abstraer a los clientes de los detalles de la comunicación con los robots que componen el SMR. Los clientes, normalmente, serán agentes u otros procesos software que son necesarios en las distintas aplicaciones de los SMR.

Los componentes de la capa SIAN proporcionan las interfaces que describen la funcionalidad ofrecida por los robots (objetos dispositivos). Las interacciones entre los robots y los clientes son representadas por "proxies", los cuales encapsulan la información necesaria para acceder a los robots. El framework propuesto utiliza un middleware orientado a objetos como CORBA o ZeroC ICE para el soporte de las comunicaciones entre los componentes.

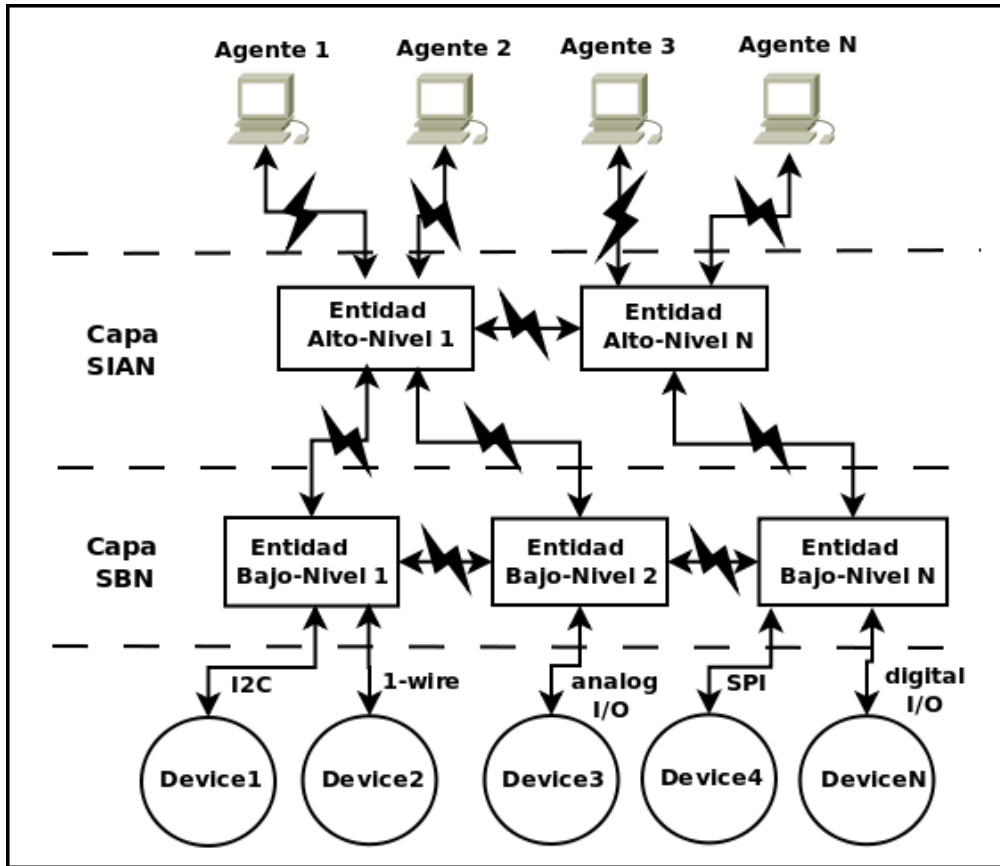


Figura 3.1: Visión general de la arquitectura propuesta.

Además, esta propuesta utiliza un servicio de publicación/subscripción para comunicar a los clientes u otros robots los cambios de estado de los componentes del SMR. Los tipos de evento son instanciados dinámicamente y permiten la publicación de información y la subscripción a un determinado tipo de noticia asegurando la escalabilidad.

El soporte multi-thread ofrecido por el middleware permite que el framework propuesto pueda atender de forma simultanea varias peticiones procedentes de los clientes. Es muy importante liberar cuanto antes el hilo de control que soporta la petición realizada, no sólo para liberar recursos, sino también para que los procesos de la capa SIAN puedan ejecutarse de una manera no-bloqueante. Para ello, el objeto dispositivo sobre el que se realiza la petición envía un mensaje que contiene la petición realizada a la capa SBN junto con un identificador que permitirá asociar la respuesta obtenida de la capa SBN al hilo de control encargado de enviar la respuesta al cliente que realizó dicha petición; esta asociación es almacenada en un diccionario interno.

Después del proceso anterior, los mensajes serán almacenados en colas (hay

una cola por cada componente de la capa de SBN) que permitirán liberar los hilos de control manejados por el middleware para el tratamiento de las peticiones. Mientras tanto, el cliente puede optar por bloquearse a la espera de una respuesta o por utilizar un objeto *call-back*, permitiendo así que la capa SIAN le envíe la respuesta cuando ésta esté disponible. Cada cola es mantenida por un hilo de control que se encargará de ir enviando el mensaje almacenado (encolado) previamente al componente correspondiente de la capa SBN. Cuando la petición es resuelta, el componente de la capa SBN genera un mensaje de respuesta etiquetado con el mismo identificador utilizado en el mensaje de petición enviado anteriormente, esto permite asociar las peticiones con las respuestas obtenidas. Una vez que la capa SIAN recibe la respuesta del mensaje, identifica al cliente que realizó la petición usando el diccionario, formatea los datos recibidos de acuerdo con la operación de la interfaz invocada por el cliente y por último, envía los datos de vuelta al cliente.

Este mecanismo puede ser utilizado para tratar las peticiones explícitas de los clientes. Sin embargo, como se ha comentado anteriormente, existe otro tipo de interacciones. Por ejemplo, un cliente puede obtener notificaciones de ciertos dispositivos, sin necesidad de realizar una petición previa. Así, los robots no se comportan como meros esclavos, ya que son capaces de enviar notificaciones acerca de los cambios de estado de los sensores, actuadores o cambios en el comportamiento de éstos. Para ello, un proceso SIAN utiliza el servicio de publicación/subscripción para definir tipos de eventos (noticias) asociados a los dispositivos cuyos estados se notificarán a los clientes, que deseen suscribirse a estas noticias. Por lo tanto, cuando un cliente solicita cambios en el estado de un dispositivo, estos cambios también se publican para que los clientes suscritos puedan obtener esta información.

El problema de la escalabilidad también es abordado, ya que los procesos de la capa SIAN pueden ser ejecutados en diferentes plataformas. Esto posibilita elegir la infraestructura hardware y el sistema operativo más adecuado para el dominio de aplicación. Los servicios del middleware utilizado también pueden ser utilizados para integrar los mecanismos de localización y replicación dentro de la capa SIAN.

Otro aspecto muy importante, es la sincronización entre los distintos componentes de las capas SIAN y SBN. Antes de que los clientes puedan utilizar los servicios ofrecidos por los robots, se iniciará un proceso de sincronización con los distintos elementos de la capa SBN. Con esta sincronización, se podrá calcular la diferencia de tiempos en el envío y recepción de los mensajes intercambiados entre los distintos componentes de la capa SIAN y SBN.

Actualmente, se está trabajando en un servicio de despliegue que va a permitir el registro dinámico de nuevos robots en el sistema. Cuando se incorpora un nuevo robot al sistema, éste envía un mensaje a la capa SIAN para que ésta lance los procesos encargados de ofrecer el soporte necesario para que los clientes puedan acceder a la funcionalidad descrita por el nuevo robot. Teniendo en cuenta que la mayoría de sistemas multi-robot están compuestos por robots heterogéneos, los

nuevos componentes quedarán registrados en una ontología que permitirá realizar una clasificación de los robots del sistema, dependiendo de la funcionalidad ofrecida por estos. Además, utilizando el servicio de publicación/subscripción, la capa SIAN notificará a los demás robots la incorporación de un nuevo componente. Esta ontología será actualizada periódicamente mediante la consulta del estado de los distintos robots. Por último, se están implementando diferentes componentes que permitirán utilizar distintos interfaces de comunicación como Bluetooth, Zigbee, etc. para comunicar ambas capas.

3.2. Capa SBN

La capa SBN es la responsable de resolver las cuestiones de conectividad con los robots y con los dispositivos que éstos incorporan. Para ello, se ha diseñado una solución sencilla pero flexible basada en la utilización de los microcontroladores que incorporan los robots como servidores hardware distribuidos. El esquema permite que los robots inicien, modifiquen o coordinen las diferentes conductas reactivas que ejecutan, ya que la capacidad de procesamiento de los microcontroladores actuales no es suficiente para ejecutar conductas más complejas.

Por lo tanto, el framework propuesto permite el control directo de los dispositivos (sensores y actuadores) de los distintos robots o la modificación de las conductas reactivas que éstos describen. Para ello, los componentes de la capa SBN aceptarán peticiones de otros componentes y enviarán los resultados correspondientes a través de una interfaz de comunicación y un protocolo de mensajes bien definido.

Un método adecuado con el que proporcionar un acceso remoto a los dispositivos de los robots sería incluir en estos una interfaz de red, pero muy frecuentemente los robots actuales de bajo perfil (nota al pie y lo explico) no proporcionan los recursos necesarios para implementar una pila de protocolos de red. En cambio, casi todos los robots proporcionan una o varias interfaces serie para comunicarse con otros componentes, que pueden ser utilizados de forma inalámbrica por radiofrecuencia (RadioModem) si es necesario. La solución propuesta es la conversión entre el protocolo serie y el protocolo de red utilizado para conseguir así una homogeneidad de los distintos interfaces de comunicación utilizados por los componentes del SMR. Es importante señalar que, la mayoría de las veces, estos conversores están conectados a las interfaces serie en calidad de dispositivos externos. Los conversores más utilizados son los conversores de RS-232 a TCP/IP y viceversa.

Como se ha visto en el capítulo anterior, en la mayoría de sistemas multi-robot, los robots ejecutan diferentes tareas que pueden ser ordenadas por clientes externos o por otros robots. En este caso, la capa SBN ofrece el soporte necesario

para que los robots puedan recibir mensajes de notificación de clientes remotos u otros robots, mientras están ejecutando alguna tarea.

Además, como se ha comentado anteriormente, los robots pueden publicar cambios de estado sin necesidad de que exista una petición previa. En estos casos, los robots publican esta información por vía del servicio de publicación/subscripción de la capa SIAN para que esta información sea comunicada a los componentes suscritos a ese tipo de noticia. Por otra parte, los robots pueden estar suscritos a determinados tipos de noticia para obtener información de los otros robots, evitando así que un robot tenga que realizar un envío masivo de mensajes a los demás.

Por último, para realizar el proceso de sincronización con la capa SIAN se utiliza el reloj de tiempo real disponible en los microcontroladores que incorporan la mayoría de los robots. En el caso de no disponer de este recurso, deberán utilizarse dispositivos externos para poder realizar el proceso de sincronización.

3.3. Casos de Estudio

En esta sección se describen varios ejemplos de sistemas robóticos implementados con el framework de desarrollo propuesto en este trabajo fin de máster.

En la implementación de estos casos de estudio se ha decidido utilizar el middleware ZeroC Ice para dar soporte a las comunicaciones de la capa SIAN. Este middleware representa una moderna alternativa para el desarrollo de sistemas distribuidos, al disponer de características análogas a las de CORBA pero con una complejidad reducida y una gran cohesión que facilita su uso y su aprendizaje [12].

El lenguaje de descripción de interfaces definido por Ice es Slice. Éste es un lenguaje de especificación de interfaces que cumple una finalidad análoga a la de IDL en el caso de CORBA. La definición de las interfaces del framework se realizarán mediante Slice.

En la actualidad Ice proporciona *mappings* de Slice para C++, Java, C#, Visual Basic, Python, Ruby y PHP en Windows y en UNIX.

La elección de Ice ha estado también motivada por el gran número de servicios disponibles que permiten abordar diferentes cuestiones:

- **IceGrid**: Un servicio de localización y activación
- **Freeze**: Un conjunto de servicios de persistencia
- **Glacier2**: Un servicio de *firewall* para aplicaciones Ice

- **IceBox**: Un framework para la configuración de estos servicios
- **IceStorm**: Un servicio de publicación/subscripción
- **IcePatch2**: Un servicio de *patching* para aplicaciones Ice

Por último, como servicio de publicación/subscripción se ha utilizado el servicio IceStorm que utiliza sus entidades características, llamadas *topics*, para comunicar a los clientes u otros robots los cambios de estado de los componentes del SMR. Los *topics* son instanciados dinámicamente y permiten la publicación de información y la subscripción a un determinado tipo de noticia asegurando la escalabilidad.

3.3.1. Control de un robot móvil autónomo

En el Grupo de Investigación Oreto de la Universidad de Castilla-La Mancha esta arquitectura ha sido utilizada para controlar de forma remota un vehículo articulado construido en el propio laboratorio con dos motores de CC para manejar el sistema de tracción, dos servos para controlar el movimiento de una cabeza móvil con dos grados de libertad, diodos LED para emitir señales de luminosas y un *buzzer* piezoeléctrico para emitir señales acústicas. El vehículo también incluye sensores de ultrasonido e infrarrojos para la detección de obstáculos, una brújula digital, un termómetro digital para medir la temperatura del interior de la plataforma y en un sensor PIR para la detección de movimiento. La Figura 3.2 muestra una imagen de esta plataforma.

Para implementar la capa SBN se ha utilizado el microcontrolador **BasicX-24p**®¹ con una interfaz UART para la comunicación serie, conectada al dispositivo **SitePlayer Telnet**®² que permite realizar la conversión entre un puerto serie y un puerto TCP/IP. Para obtener una conectividad inalámbrica, se ha utilizado un router inalámbrico (tipo **Fonera**³) configurado como punto de acceso y conectado al SitePlayer Telnet. La capa SIAN reside en una máquina externa con conectividad inalámbrica para comunicarse con el anterior router. La Figura 3.3 muestra un esquema de la estructura de comunicaciones de los dispositivos de la capa SBN.

Finalmente se ha desarrollado una aplicación cliente que permite la teleoperación y obtención remota de la información proporcionada por los sensores de la plataforma con resultados plenamente satisfactorios.

¹<http://www.basicx.com/>

²<http://www.netmedia.com/siteplayer/telnet/index.html>

³<http://www.fon.com/>



Figura 3.2: Robot Móvil Autónomo ATOS I.

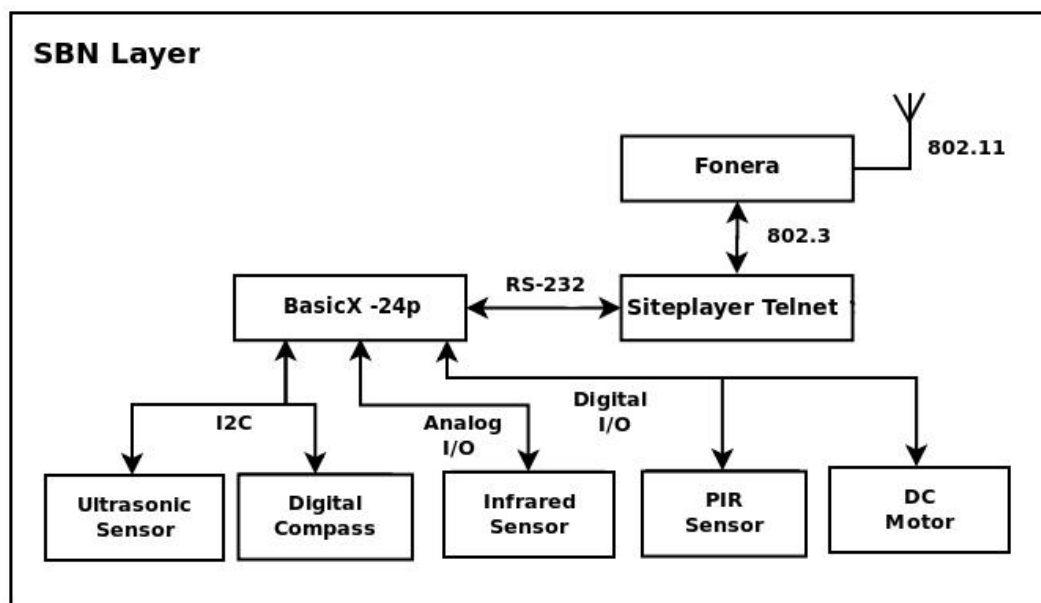


Figura 3.3: Esquema de conexiones de los dispositivos en la capa SBN.

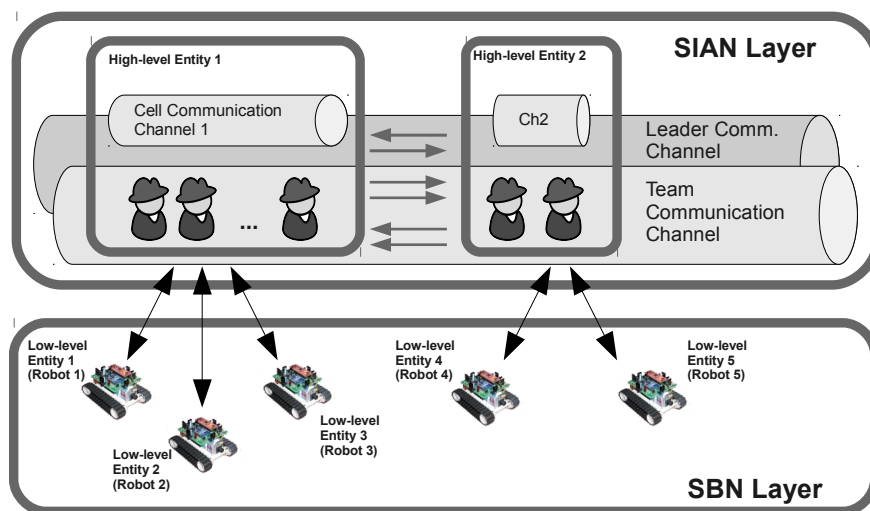


Figura 3.4: Utilización del framework propuesto en una arquitectura multi-agente.

3.3.2. Coordinación de un sistema multi-robot

Los SMR, en muchos de los casos, necesitan de un soporte externo ofrecido por agentes para distintas tareas. En este caso de estudio se ha implementado un sistema multi-robot con soporte multi-agente que permite la coordinación de diferentes plataformas homogéneas. En particular, se ha implementado una tarea de vigilancia, aprovechando el soporte multi-agente implementado en el trabajo de Vallejo et al. [34], donde a cada robot se le asigna un rol que determina el canal de comunicaciones del servicio de publicación/subscripción que puede utilizar para publicar la información recogida por sus sensores. La Figura 3.4 muestra un diagrama de la arquitectura multi-agente propuesta.

En este caso, la aplicación del framework ha sido inmediata, ya que permite una comunicación bidireccional entre el soporte multi-agente y los diferentes componentes (plataformas robóticas) de la capa SBN.

Los componentes de la capa SBN son implementados en los microprocesadores de las distintas plataformas robóticas. En nuestro caso, se han utilizado varios robots **Surveyor SRV-1 Blackfin Robot**⁴. Estos robots disponen de una cámara digital, una interfaz de comunicación inalámbrica 802.11b/g, dos diodos láser y cuatro sensores de ultrasonido para la detección de obstáculos. Al igual que en el caso de estudio anterior, la capa SIAN reside en una máquina externa con una interfaz inalámbrica.

⁴<http://www.surveyor.com/>

Capítulo 4

Conclusiones y Líneas de Investigación Futuras

4.1. Conclusiones

En este trabajo de investigación se ha propuesto un framework de desarrollo para sistemas multi-robot que permite el acceso remoto a dispositivos de entrada/salida.

Considerando el estado del arte revisado en la capítulo 2, la mayoría de arquitecturas estudiadas proponen distintas soluciones para dar soporte a la comunicación entre los robots y agentes. Como ya se discutió en dicho capítulo, los problemas más comunes de estas arquitecturas están relacionados con aspectos como el acceso a los robots, la incorporación de nuevos robots, la tolerancia a fallos y la latencia en las comunicaciones. Aunque se proponen diferentes soluciones para resolver estos problemas, la mayoría de arquitecturas no consideran estos aspectos debido a que son muy específicas y están centradas en otros aspectos, u ofrecen soluciones muy genéricas cuya aplicación al dominio de los SMR no es trivial.

Con el objetivo de tratar esta problemática existente en la literatura, en este trabajo se ha propuesto un framework específico para el desarrollo de sistemas multi-robot que ofrezca el soporte de comunicaciones necesario para desarrollar cualquier tipo de sistema multi-robot, utilizando robots homogéneos o heterogéneos pero que al mismo tiempo, pueda ser utilizado al mismo tiempo, para desarrollar otro tipo de sistemas reactivos.

Además, se han propuesto diferentes mecanismos y técnicas para tratar las siguientes cuestiones:

- **Latencia en las comunicaciones:** En este trabajo se propone la utilización de un protocolo de mensajes que evite el envío de grandes volúmenes de

información. Además, cada tipo de robot se gestiona independientemente en la capa de alto nivel. De este modo, permite que no haya conflictos en la comunicación con los demás dispositivos y que con la sincronización de ambas capas (SIAN y SBN) se pueda controlar el flujo de información, identificando los posibles retrasos en los mensajes recibidos.

- **Tolerancia a fallos:** Este framework permite la replicación de los componentes de la capa SIAN, incrementando la robustez del sistema a la hora de desplegar el SMR, evitando así los posibles errores en la comunicación con los agentes. Además, mediante la utilización de una ontología para clasificar los robots del SMR, en función del tipo de servicio, se evitan los posibles fallos relacionados con los robots. Si un robot fallase, el sistema podría reasignar la tarea que ese robot tuviera asignada a otro robot similar observando la ontología.
- **Facilidad a la hora de integrar nuevos dispositivos:** El servicio de despliegue propuesto en este framework permite un registro dinámico de nuevos robots en el sistema y, utilizando el servicio de publicación/subscripción, notificar dicho registro a todos los componentes del SMR.

Con la utilización del servicio de publicación/subscripción se permite la creación de distintos canales de eventos. Gracias a ellos, los robots o agentes pueden publicar distintos tipos de noticias sin necesidad de conocer el número de robots o agentes desplegados ni la ubicación de los mismos.

Por último, se han descrito dos casos de estudio que permiten mostrar el uso del framework propuesto en distintos ámbitos de aplicación. En el primer caso de estudio se describe un sistema con único robot, en el cuál se han tratado de solucionar aspectos como el acceso a los dispositivos y la latencia en las comunicaciones. En el segundo caso se muestra el uso del framework propuesto en un sistema multi-robot, con el fin de solucionar aspectos como el acceso a los robots por parte de una capa multi-agente, la tolerancia a fallos o la incorporación de nuevos robots al SMR.

4.2. Líneas futuras de investigación

A continuación se exponen las líneas de investigación más relevantes que se abren tras la realización del presente trabajo.

- Integración en el framework de un lenguaje que permita modelar el dominio de aplicación de los SMR, creando el soporte gráfico necesario para realizar

el diseño e implementación de un SMR de una forma sencilla. Por ejemplo, el entorno de programación de los robots Moway¹.

- Estudio detallado de protocolos seguros que eviten intrusiones no deseadas como, por ejemplo, que determinados agentes utilicen ciertos dispositivos.
- Integración de un mecanismo de notificación múltiple a partir de una descripción semántica. De este modo, un dispositivo pueda publicarse en multiples proxies, consiguiendo así, que dependiendo de los cambios de estado de un dispositivo, pueda publicarse una información u otra.
- Definición de dispositivos complejos (grupos de robots) que deleguen en dispositivos simples, es decir, establecer un mecanismo de creación de dispositivos complejos mediante la composición de dispositivos simples [15].
- Estudio e integración de ontologías de robots que permitan que los potenciales clientes puedan conocer las características de los objetos de servicio.
- Integración y estudio de mecanismos de coordinación que permitan el desarrollo de distintos tipos de SMR.
- Despliegue automático a partir de descripciones semánticas de los componentes del SMR.
- Asignación dinámica de tareas que permitan diseñar sistemas multi-robot con distintos comportamientos [7].

¹<http://www.moway-robot.com/>

Bibliografía

- [1] T. Arai, E. Pagello, and L. E. Parker. Guest editorial advances in multirobot systems. *IEEE Transactions on Robotics and Automation*, 18(5):655–661, 2002.
- [2] T. Balch. The Impact of Diversity on Performance in Multi-robot Foraging. In *In Proc. Autonomous Agents 99*, pages 92–99. ACM Press, 1999.
- [3] T. Balch and A. Ram. Integrating robotics research with JavaBots. In *Working Notes of the AAAI Spring Symposium*, 1998.
- [4] C. Castelpietra, L. Iocchi, D. Nardi, M. Piaggio, A. Scalzo, and A. Sgorbissa. Coordination among heterogenous robotic soccer players. In *Proc. of International Conference on Intelligent Robots and Systems (IROS'2000)*, volume 2, pages 1385–1390, 2000.
- [5] E. Domínguez, J.C. Hidalgo, and L. Jiménez. ATOS: Una Arquitectura Basada en Ice para el Acceso Distribuido a Dispositivos de Entrada/Salida. In *XIII Workshop en Sistemas Distribuidos y Paralelismo (WSDP), Jornadas Chilenas de Computación*, 2009.
- [6] B. Gerkey, R. T. Vaughan, and A. Howard. The Player/Stage Project: Tools for Multi-Robot and Distributed Sensor Systems. In *Proceedings of the 11th International Conference on Advanced Robotics*, pages 317–323, 2003.
- [7] B.P. Gerkey and M.J. Mataric. Sold!: auction methods for multirobot coordination. *IEEE Transactions on Robotics and Automation*, 18(5):758–768, oct. 2002.
- [8] M. Hannebauer, J. Wendler, E. Pagello, L. Iocchi, D. Nardi, and M. Salerno. Reactivity and Deliberation: A Survey on Multi-Robot Systems. In *Balancing Reactivity and Social Deliberation in Multi-Agent Systems*, volume 2103 of *Lecture Notes in Computer Science*, pages 9–32. Springer Berlin / Heidelberg, 2001.

- [9] D. Harel, H. Lachover, A. Naamad, A. Pnueli, M. Politi, R. Sherman, and a. Shtul-Trauring. STATEMATE: A Working Environment for the Development of Complex Reactive Systems. In *ICSE '88: Proceedings of the 10th international conference on Software engineering*, pages 396–406, 1988.
- [10] D. Harel and A. Pnueli. On the Development of Reactive Systems. *Logics and models of concurrent systems*, pages 477–498, 1985.
- [11] M. Henning. A New Approach to Object-Oriented Middleware. *IEEE Internet Computing*, 8(1):66–75, 2004.
- [12] M. Henning. The rise and fall of CORBA. *Communications of the ACM*, 51(8):52–57, 2008.
- [13] M. Henning and S. Vinoski. *Advanced CORBA Programming with C++*. Addison Wesley, 1999.
- [14] L. Hugues. Grounded Representations for a Robots Team. In *Proceedings of IROS 2000, IEEE/RSJ Intern. Conference on Intelligent Robots and Systems*, volume 3, pages 2248–2253, 2000.
- [15] M.N. Huhns and M.P. Singh. Service-oriented computing: key concepts and principles. *IEEE Internet Computing*, 9(1):75–81, 2005.
- [16] D. Jung and A. Zelinsky. Grounded Symbolic Communication between Heterogeneous Cooperating Robots. *Autonomous Robots*, 8(3):269–292, 2000.
- [17] C. Kube and Eric Bonabeau. Cooperative transport by ants and robots. *Robotics and Autonomous Systems*, 30(1-2):85–101, January 2000.
- [18] S. Magnenat, P. Rétornaz, M. Bonani, V. Longchamp, and F. Mondada. ASE-BA: A Modular Architecture for Event-Based Control of Complex Robots. *IEEE/ASME Transactions on Mechatronics*, (99):1–9, 2010.
- [19] M. Montemerlo, N. Roy, and S. Thrun. Perspectives on standardization in mobile robot programming: The carnegie mellon navigation (CARMEN) toolkit. In *In Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, pages 2436–2441, 2003.
- [20] L. Mota and L. Reis. A Common Framework for Co-operative Robotics: An Open, Fault Tolerant Architecture for Multi-league RoboCup Teams. In *Simulation, Modeling, and Programming for Autonomous Robots*, volume 5325 of *Lecture Notes in Computer Science*, pages 171–182. Springer Berlin / Heidelberg, 2008.

-
- [21] F. Moya, D. Villa, F. J. Villanueva, J. Barba, F. Rincón, and J. C. López. Embedding standard distributed object-oriented middlewares in wireless sensor networks. *Wireless Communications and Mobile Computing*, 9(3):335–345, 2009.
 - [22] F. R. Noreils. Toward a robot architecture integrating cooperation between mobile robots: application to indoor environment. *International Journal of Robotics Research*, 12(1):79–98, 1993.
 - [23] A. Orebäck. *A Component Framework for Autonomous Mobile Robots*. PhD thesis, KTH Numerical Analysis and Computer Science, Stockholm, 2004.
 - [24] A. Orebäck and H.I. Christensen. Evaluation of Architectures for Mobile Robotics. *Autonomous Robots*, 14(1):33–49, 2003.
 - [25] L. E. Parker. ALLIANCE: An Architecture for Fault Tolerant Multi-Robot Cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, 1998.
 - [26] L. E. Parker. Cooperative robotics for multi-target observation. *Intelligent Automation and Soft Computing*, 5(1):5–19, 1999.
 - [27] P. Petersson, D. Austin, and H.I. Christensen. Dca: A distributed control architecture for robotics. In *the IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems (IROS)*, pages 2361–2368, October 2001.
 - [28] J. L. Posadas, J. L. Poza, J. E. Simó, G. Benet, and F. Blanes. Agent-based distributed architecture for mobile robot control. *Engineering Applications of Artificial Intelligence*, 21(6):805–823, 2008.
 - [29] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng. ROS: an open-source Robot Operating System. In *ICRA Workshop on Open Source Software*, 2009.
 - [30] A. Siegberg, A. Bredenfeld, H. Guenther, and et al. Team Description of the GMD RoboCup-Team. In *RoboCup-98: Robot Soccer World Cup II*, volume 1604 of *Lecture Notes in Computer Science*, pages 473–478. Springer Berlin / Heidelberg, 2010.
 - [31] R. Simmons, D. Apfelbaum, D. Fox, R. P. Goldman, K. Z. Haigh, D. J. Musliner, M. Pelican, and S. Thrun. Coordinated Deployment of Multiple, Heterogeneous Robots. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2254–2260, 2000.
 - [32] C. Szyperski, D. Gruntz, and S. Murer. *Component Software: Beyond Object-Oriented Programming*. Addison-Wesley, 2st edition, 2002.

- [33] H. Utz, S. Sablatnog, S. Enderle, and G. Kraetzschmar. Miro - middleware for mobile robot applications. *IEEE Transactions on Robotics and Automation*, 18(4):493–497, 2002.
- [34] D. Vallejo, P. Remagnino, D. N. Monekosso, L. Jiménez, and C. González. A Multi-agent Architecture for Multi-robot Surveillance. In *ICCCI '09: Proceedings of the 1st International Conference on Computational Collective Intelligence. Semantic Web, Social Networks and Multiagent Systems*, pages 266–278, 2009.
- [35] M. Veloso and P. Stone. Individual and Collaborative Behaviors in a Team of Robotic Soccer Agents. In *Proceedings of International Conference on Multi Agent Systems*, pages 309–316, 1998.
- [36] Y. Wang and C. W. de Silva. A machine-learning approach to multi-robot coordination. *Engineering Applications of Artificial Intelligence*, 21(3):470–484, 2008.
- [37] G. Weiss. *Multiagent systems: a modern approach to distributed artificial intelligence*. MIT Press, Cambridge, MA, USA, 1999.
- [38] B. Werger and M. J. Mataric. Broadcast of Local Eligibility for Multi-Target Observation. In *Proceedings Conference on Distributed Autonomous Robotic Systems*, pages 347–356, 2000.
- [39] M. Wooldridge. *Introduction to MultiAgent Systems*. John Wiley & Sons, Inc. New York, second edition, 2009.
- [40] T. Yang, J. Ma, Z. G. Hou, G. Peng, and M. Tan. A Multi-agent Architecture Based Cooperation and Intelligent Decision Making Method for Multirobot Systems. *Neural Information Processing: 14th International Conference, ICONIP 2007*, pages 376–385, 2008.
- [41] K. Yokota, K. Ozaki, N. Watanabe, A. Matsumoto, D. Koyama, T. Ishikawa, Kuniaki Kawabata, Hayato Kaetsu, and Hajime Asama. UTTORI United: Cooperative Team Play Based on Communication. In *RoboCup-98: Robot Soccer World Cup II*, pages 479–484, 1999.

Apéndice A

Resumen de las asignaturas cursadas en el Máster

A.1. Redes Multimedia

Título: Redes Multimedia.

Profesorado: Dr. Antonio Garrido del Solo, Dr. Pedro Cuenca Castillo, Dr. Francisco Manuel Delicado Martinez, Dra. Teresa Olivares Montes, Dr. Gerardo Fernandez Escribano, Dr. José Miguel Villalón Millán y Dr. José Luis Martinez Martinez.

Contenido:

El objetivo principal de la asignatura es enseñar al alumno las diferentes tecnologías de red con soporte de QoS, especialmente diseñadas para cubrir las necesidades de las aplicaciones Multimedia, fundamentalmente para la transmisión de vídeo y audio.

Los métodos de enseñanza aplicados en la asignatura son básicamente tres: clases magistrales impartidas por los profesores citados anteriormente, lectura de artículos científicos y debate entre los alumnos sobre dichos artículos. En cuanto al método de evaluación se tiene en cuenta la asistencia y la participación en clase, así como la entrega de uno o varios trabajos individuales y/o en grupo, que siempre debe ir acompañada de una exposición oral al resto de alumnos.

Resultados:

La primera actividad consistía en completar un cuestionario relacionado con la QoS en las BWA. Posteriormente se realizó un trabajo de "Introducción a la Transcodificación de Vídeo" donde se realizó un comentario de la codificación de video H.264. Por último, se realizaron dos comentarios de texto de los siguientes artículos "iPower: An Energy Conservation System for Intelligent Buildings by Wireless

Sensor Networks” del campo de las WSN y ”Developments and Trends on Video Coding: is there a xVC Virus?” del campo de la codificación de video.

A.2. Sistemas distribuidos avanzados: grid e inteligencia ambiental

Título: Sistemas distribuidos avanzados: grid e inteligencia ambiental.

Profesorado: Dra. Camelia Muñoz Caro, Dr. Alfonso Niño Ramos y Dr. José Bravo Rodríguez.

Contenido:

La asignatura se divide en dos partes bien diferenciadas. Una primera parte titulada Sistemas Grid, impartida por los profesores Camelia Muñoz y Alfonso Niño. Y una segunda parte titulada inteligencia ambiental impartida por el profesor José Bravo. La primera de ellas tiene como gran objetivo dotar al alumno de los conocimientos necesarios para configurar y gestionar infraestructuras grid, así como diseñar sistemas grid a nivel de servicios básicos y servicios web. Respecto a la segunda parte, se estudian nuevas formas de interacción en donde se manifiesta la proactividad de los sistemas. Además, se pretende capacitar al alumno para el modelado de acciones de usuario en sistemas Aml, realizándose estudios reales sobre instalaciones en diferentes contextos, entre los que cabe destacar los centros de enseñanza y hospitalarios.

El método de evaluación de la asignatura se basa en la valoración de la asistencia a clase, participación en las actividades propuestas y, finalmente, en la entrega y defensa de dos trabajos teóricos.

Resultados:

Para superar la asignatura se realizaron y defendieron dos trabajos teóricos:

- *Computación GRID y su aplicación en el campo de la Astronomía.* En este trabajo se realizó una introducción a los sistemas Grid analizando su evolución histórica y aquellos elementos que lo distinguen de los sistemas distribuidos tradicionales. La parte central del documento está enfocado a las aplicaciones de Grid en el campo de la astronomía.
- *Grid computing vs P2P.* En este trabajo se describieron las similitudes y diferencias existentes entre dichos paradigmas.

A.3. Sistemas heterogéneos en red

Título: Sistemas heterogéneos en red.

Profesorado: Dr. Juan Carlos López López, Dr. Francisco Moya Fernández, Dr. David Villa Alises, Dr. Félix Jesús Villanueva Molina , Dr. Jesús Barba Romero y Dr. Fernando Rincón Calle.

Contenido:

En esta asignatura se estudiaron distintos aspectos relacionados con la interconexión de sistemas complejos tanto a nivel hardware como a nivel software. Por otra parte, también se analizan distintas metodologías de desarrollo para los sistemas empujados como punto de partida para la integración de sistemas completos dentro de un chip, es decir, desde los denominados system-on-a-chip hasta la integración de redes en el propio chip, o redes en chip.

El método de evaluación de la asignatura se basa en la valoración de la asistencia a clase, participación en las actividades propuestas y, finalmente, en la entrega y defensa de un trabajo teórico.

Resultados:

Para superar la asignatura se realizó el siguiente trabajo "Modificación de la arquitectura ATOS para sistemas empujados de bajo coste". En este trabajo se realizó una modificación del framework ATOS para empujarlo en los dispositivos de bajas prestaciones como por ejemplo, un router inalámbrico tipo La Fonera.

A.4. Pruebas y seguridad de Sistemas de Información

Título: Pruebas y seguridad de Sistemas de Información.

Profesorado: Dr. Eduardo Fernández Medina Patón y Dr. Macario Polo Usaola.

Contenido:

En el primer bloque de la asignatura, se aborda la problemática de la seguridad en los Sistemas de Información. Dado que la seguridad constituye un problema transversal, se estudian las diferentes dimensiones de la seguridad avanzada abordando desde la criptografía hasta las bases de datos pasando por la seguridad en procesos de negocio, almacenes de datos y servicios web.

Resultados:

Para esta parte de la asignatura, se analizó el artículo "A common criteria based security requirements engineering process for the development of secure information systems" y se escribió un trabajo relacionado con la seguridad en las redes

sociales. En dicho trabajo se describen algunos elementos importantes relacionados con la seguridad en el acceso y la utilización de dichas redes.

El segundo bloque de la asignatura esta orientado a las pruebas de Sistemas de Información centrandose en la adquisición de destrezas y conocimientos relacionados con los principales métodos y técnicas de las pruebas del software. Se estudian los distintos niveles de prueba: de caja negra, de caja blanca, unitarias, de integración y de sistema. También se profundiza en el **análisis de las medidas de cobertura** en las pruebas de caja blanca y en las estrategias de combinacion para la generación de casos de prueba.

Resultados:

Se realizo un trabajo donde dado un conjunto de SUTs, se determinó justificadamente un conjunto de valores de prueba utilizando una o más técnicas (clases de equivalencia, valores límite, etc.). Después se combinaron los valores obtenidos con un algoritmo de combinación (sistema CTWeb), obteniendo así un test suite TS y por último, se ejecuto el TS anterior contra cada SUT utilizando los diversos modos de ejecución de Bacterio (Weak Mutation, Flexible Weak Mutation, Functional Qualification).

A.5. Cognición y colaboración

Título: Cognición y Colaboración.

Profesorado: Dr. José Ángel Olivas Varela, Dr. Manuel Prieto Méndez y Dra. Carmen Lacave Rodero.

Contenido: La asignatura está estructurada en tres bloques temáticos bien diferenciados, cada uno de ellos impartido por uno de los profesores de la asignatura.

- Bloque I: Gestión del conocimiento y recuperación eficiente de la información en la Web mediante técnicas de Soft Computing (impartido por el Dr. José Ángel Olivas). El principal objetivo de este primer bloque es enseñar al alumno las principales técnicas basadas en Soft-Computing (tolerantes a la imprecisión e incertidumbre) para manipular información y extraer conocimiento de diversas fuentes.
- Bloque II: Gestión del conocimiento y computación gráfica (impartido por la Dra. Carmen Lacave Rodero). En este bloque se realiza una introducción al razonamiento probabilístico, haciendo mayor hincapié en el razonamiento con Redes Bayesianas.

- Bloque III: Gestión del conocimiento en grupo y aprendizaje comparado (impartido por el Dr. Manuel Prieto Méndez). El principal objetivo de este bloque es conocer las taxonomías y principios de funcionamiento de los sistemas de trabajo en grupo: análisis de los modelos CSCW, patrones para trabajo en grupo, aprendizaje automático y aprendizaje humano, etc.

Para superar el curso, el alumno debe realizar un trabajo encuadrado en alguno de los tres bloques anteriores.

Resultados: Se realizó una serie de ejercicios relacionados con los distintos aspectos de las Redes Bayesianas corregidos por la Dra. Carmen Lacave Roderó.

A.6. Técnicas de softcomputing

Título: Técnicas de softcomputing.

Profesorado: Dr. Luis Jiménez Linares, Dr. José Jesús Castro Sánchez y Dr. Luis Rodríguez-Benítez.

Contenido:

En esta asignatura se presenta al alumno una nueva perspectiva para la definición de sistemas complejos mediante sistemas difusos, diferente a los métodos clásicos aprendidos durante la ingeniería en informática. Además, se pretende que el alumno obtenga cierta destreza en el uso y manejo de sistemas de reglas difusas, así como en el manejo de distintas metodologías para la definición de sistemas difusos. Para superar la asignatura se valora positivamente la asistencia a clase, así como la realización y la defensa de dos trabajos teórico-prácticos encuadrados en alguno de los temas principales enumerados anteriormente.

Resultados:

Se realizaron dos trabajos:

- Un trabajo teórico titulado “La Lógica Difusa en el campo de la Robótica”. En el trabajo se realizó una introducción a la lógica difusa, así como a los métodos de inferencia. En la segunda parte se describía las distintas aplicaciones basadas en lógica difusa utilizadas en el campo de la robótica .
- Trabajo teórico-práctico en el que se realizó la implementación de un control difuso en microcontroladores de bajo coste.

Apéndice B

Curriculum Vitae

DATOS PERSONALES

- APELLIDOS: **Hidalgo Nevado**
- NOMBRE: **José Carlos**
- DNI: **5685702-X**
- FECHA DE NACIMIENTO: **01-10-1980**
- DIRECCIÓN PARTICULAR: **C/Severo Ochoa N°8, Bloque II 2°C, Ciudad Real**
- TELÉFONO: **630608040**

SITUACIÓN PROFESIONAL ACTUAL

- Contrato laboral de apoyo a la investigación de la UCLM con Ref. 2008 COB-2492 en el cual se han desarrollado trabajos relacionados con los distintos proyectos realizados en el grupo de investigación ORETO de la Escuela Superior de Informática de Ciudad Real.

FORMACIÓN ACADÉMICA

- 1999-2001: Técnico Superior en Desarrollo de Productos Electrónicos.
- 2001-2006: Ingeniero Técnico en Informática de Sistemas por la Universidad de Castilla La-Mancha

- 2006-2009: Ingeniero en Informática por la Universidad de Castilla La-Mancha.
- 2009-2010: En la actualidad estoy cursando el Máster de Tecnologías Informáticas Avanzadas de la Universidad de Castilla-La Mancha en la Escuela Superior de Informática de Ciudad Real.

MÉRITOS ACADÉMICOS

- Premio INFOGLOBAL al mejor Proyecto Fin de Carrera "Una Arquitectura para la Distribución de Información Sensorial Heterogénea" dirigido por D. Eduardo Domínguez Parra, profesor titular de la Escuela Superior de Informática de Ciudad Real.

EXPERIENCIA LABORAL

- 1996-2008: Coordinador de las Escuelas Deportivas del Ayuntamiento. Los Pozuelos de Calatrava
- 2001: Prácticas en la empresa "Informática para Ingenieros" de Ciudad Real (400 Horas), en las que se realizó montaje y desmontaje de computadores, así como la configuración de los mismos y su correspondiente venta al público.
- 2007: Monitor de laboratorio en la Escuela Superior de Informática de Ciudad Real durante el primer cuatrimestre del curso 2007-2008.
- 2008: Beca de colaboración de la UCLM con Ref. 2008-BCL-2034 en la cuál se han desarrollado trabajos relacionados con el diseño e implementación de un robot móvil autónomo en el grupo de investigación ORETO de la Escuela Superior de Informática de Ciudad Real.
- 2009-Actualidad: Contrato laboral de apoyo a la investigación de la UCLM con Ref. 2008 COB-2492 en el cual se han desarrollado trabajos relacionados con los distintos proyectos realizados en el grupo de investigación ORETO de la Escuela Superior de Informática de Ciudad Real.

CURSOS DE FORMACIÓN DOCENTE Y OTROS CURSOS RECIBIDOS

- 2003: Curso de Iteración en la Web (24 horas). Universidad de Castilla La Mancha.
- 2004: Curso de Desarrollo de Videojuegos 3D Multiplataforma (20Horas). Universidad de Castilla La Mancha.

- 2005: Seminario de Protección de Datos (20 Horas). Universidad de Castilla La Mancha.
- 2006: Obtención del carnet de Instalador Electricista. Realización de un Trabajo Académicamente Dirigido con el profesor Dr. Jesús Salido Tercero titulado "Control de un Robot Móvil Educativo mediante un microcontrolador PIC".

PUBLICACIONES

- 2009: ATOS: Una Arquitectura Basada en Ice para el Acceso Distribuido a Dispositivos de Entrada/Salida. Domínguez, E., Hidalgo, J.C. y Jiménez, L.. Jornadas Chilenas de Computación 2009.