

Un diseño simple orientado a objetos de un equipo de fútbol de robots

Martínez, Agustín Rafael. Park, Diego. Burella, Juan Matías.
Viscuso, Germán. Holik, Federico. Fortín, Sebastián.

Resumen: Se propone un diseño simple orientado a objetos de un equipo de fútbol de robots. Se distinguen en él los objetos del dominio de estudio, los mensajes que saben responder y la forma en que colaboran entre sí. A la hora de definir las actividades del robot, se sugiere un modelo cercano a la forma en que las personas se organizan para jugar al fútbol. Finalmente se detalla un caso de estudio del diseño propuesto.

Palabras clave: Fútbol-de-Robots. Diseño. Objeto. Clase.

I. Introducción

El diseño busca permitir la convivencia de mecanismos diferentes al momento de decir qué debe hacer cada jugador del equipo. Busca por otra parte servir de punto de partida o referencia para quienes intentan implementar estrategias de juego en este dominio de estudio con programación orientada a objetos.

El diseño que se presenta no es un estudio terminado del dominio del fútbol de robots y está basado en robots con dos ruedas laterales de movimiento independiente.

Una desventaja del enfoque seguido es que presupone un poder de cómputo que no siempre está dentro de los recursos de los robots con los que se trabaja en la actualidad.

El documento se vale de diagramas de instancias, de secuencias y de clases para describir el diseño.

Los diagramas de instancias tienen dos relaciones posibles entre entidades. Cada flecha denota el conocimiento por parte del objeto de donde parte, de uno del tipo que la recibe; y si la flecha es doble, el conocimiento es de varios objetos de ese tipo. Esta relación descripta por la flecha puede llevar un nombre. Los diagramas de secuencia muestran cómo los objetos colaboran en el tiempo. En todos los casos qué mensajes se desencadenan con la recepción del primero y qué objetos van apareciendo como respuesta, hasta la respuesta

final del mensaje desencadenador –de responderse nada no se indica respuesta.

Los diagramas de clases muestran los mensajes que pueden recibir los objetos que son instancias de ellas y las relaciones de herencia entre las mismas.

En todos los casos los nombres de los diagramas están en idioma inglés para mantener una compatibilidad lingüística con los tipos predefinidos, o reutilizados de otros diseños similares, del lenguaje con el que se elige trabajar.

II. Objetos del dominio de estudio

Los principales actores del dominio de estudio son los robots, de los cuales se distinguen dos tipos: aquellos que pertenecen al equipo y aquellos que son oponentes a él. El nombre que se eligió para el primer tipo es *TeamRobot* y para el segundo *OpponentRobot*. Otro actor importante en el juego de fútbol de robots, aunque pasivo, es la pelota y está representado por el tipo *Ball*.

Los tres tipos de actores comparten características: todos ellos tienen una posición y una velocidad en el campo de juego. En particular los robots tienen también una orientación. Sin embargo, un robot del equipo tiene otros elementos que lo componen y son los que determinan su comportamiento; ellos son sus actuadores, dos ruedas laterales, y su

actividad, la cual determina qué hacer en cada momento.

Otro concepto fundamental que emerge de este estudio es el de equipo; todo miembro de equipo conoce, y por ello comparte con los demás compañeros, un objeto que lo modela del tipo *Team* (ver figura 1).

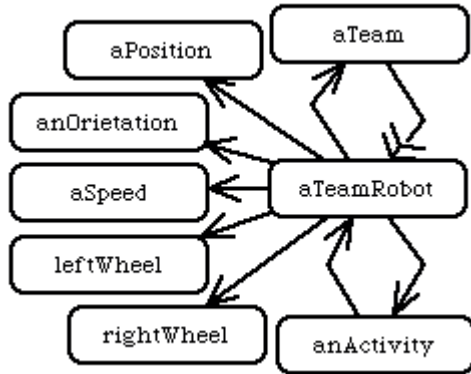


Figura 1. Diagrama de Instancias. Relaciones de conocimientos entre un objeto del tipo *TeamRobot* y sus colaboradores.

Todo robot sabe dar un paso y lo hace cada vez que recibe el mensaje *step*. Al recibir este mensaje colabora con su actividad enviando un mismo mensaje (ver figura 2). Cuando recibe este estímulo, estará determinado por las características de los individuos y la naturaleza del ambiente: el tiempo de actualización de sus sensores, o bien el de un ciclo de simulación o el de captura de una cámara de video.

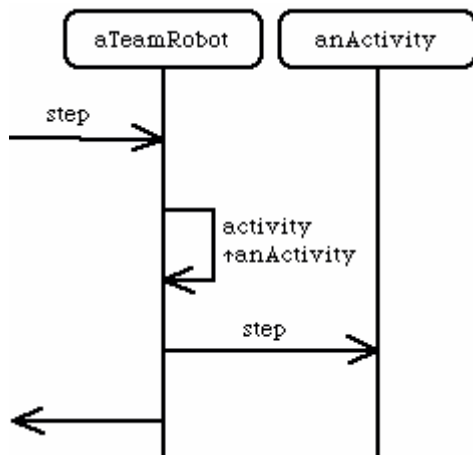


Figura 2. Diagrama de Secuencias. El paso de un robot del equipo.

Se propone modelar estos tres tipos de actores con las clases que se muestran en la figura 3.

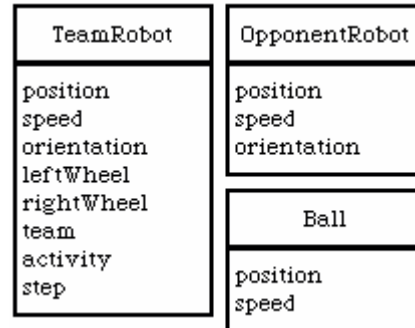


Figura 3. Diagrama de las clases de actores.

Como se describió, existe un único objeto del tipo *Team*, conocido por todo *TeamRobot*, que engloba las características del equipo. Es a través de éste que todo robot del equipo conoce a los demás miembros, los oponentes, la pelota y las diferentes áreas del campo de juego relativas a su equipo. Esto último a través de un objeto del tipo *FieldPerspective*, útil cuando el sensado entrega posiciones absolutas del campo de juego (ver figura 4).

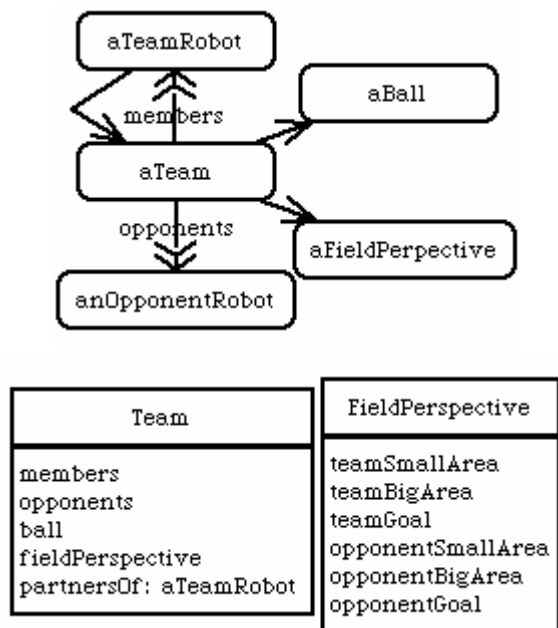


Figura 4. Diagrama de instancias y de clases. El equipo y sus colaboradores.

III. Actividades de un Robot de Fútbol

Lo expresado hasta ahora corresponde a

conceptos objetivos del dominio del fútbol de robots. Lo concerniente a la actividad de los mismos corresponde a un dominio de estudio subjetivo. El diseño, a partir de aquí, pretende acercarse al modo en que las personas se organizan al jugar un partido de fútbol. El enfoque es trazado por una serie de preguntas que pueden formularse a todo miembro del equipo. Ellas son: “¿Quién eres?”, “¿Qué eres?”, “¿Qué haces?” y “¿Cómo lo haces?”. A la pregunta “¿Quién eres?” la respuesta de este diseño ya fue definida y es “Un robot del equipo”. La respuesta a la pregunta “¿Qué eres?” será el rol del robot en el equipo. A la pregunta “¿Qué haces?” la respuesta será la acción que realiza el robot en ese momento según su rol. A la pregunta “¿Cómo lo haces?” la respuesta será la maniobra que está realizando el robot, necesaria para llevar a cabo la acción.

Siguiendo este enfoque, la actividad del robot será el rol que tiene en el equipo. Tendrá como subactividades un conjunto predeterminado de acciones posibles. Será una actividad compuesta y contará por ello con un mecanismo para la elección de la subactividad (una acción) a realizar en cada momento.

El mecanismo de elección considerará la situación del partido, es decir, el contexto de la jugada en curso. Cada situación (por ejemplo, la situación en que “el robot tiene la pelota y tiene espacio para llevarla”) es descrita a través de un estado. Más adelante se detallará en profundidad este concepto.

Los roles pertenecen al tipo *Role*, las acciones al tipo *Action* y los mecanismos de elección de subactividades al tipo *SubactivityElector*.

En la figura 5 puede observarse cómo es que se desencadenan las colaboraciones en las que el rol interviene cuando el robot debe dar un paso. Continuando el enfoque, las acciones son también actividades compuestas. De la misma forma que el rol, ellas deben contar con un

mecanismo para la elección de su subactividad, que será la maniobra. Y es igual a como se detalla en la figura 5 –cuando el rol interviene en el paso del robot eligiendo la acción– que la acción interviene eligiendo la maniobra pertinente para el momento.

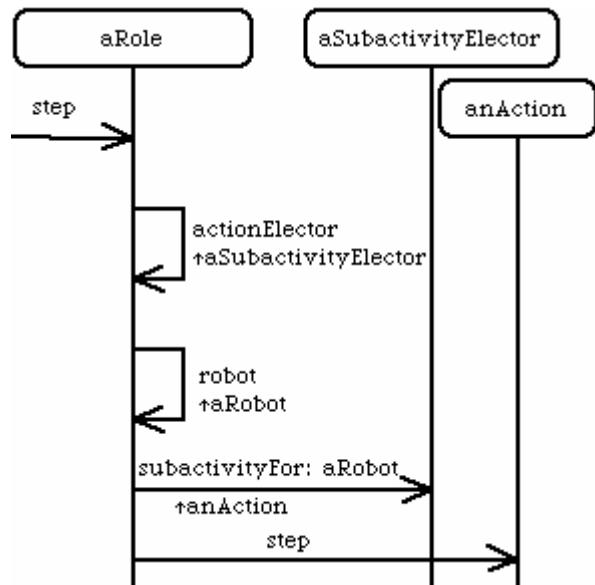


Figura 5. Diagrama de Secuencias. El paso de una actividad del tipo *Role*. Este segundo diagrama de secuencias es continuación del primero de la figura 2 (la actividad del robot es el rol).

Se proponen las clases de la figura 6 para la definición de los tipos *Role* y *Action*. Como se observa son abstractas y roles concretos heredarán de *Role* y acciones concretas heredarán de *Action*.

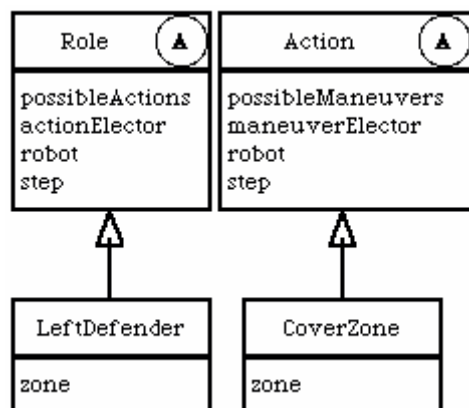


Figura 6. Diagrama de las clases abstractas *Role* y *Action* con dos subclases posibles.

Algunos roles concretos pueden ser el de *arquero*, *defensor por derecha*, *defensor por izquierda*, *centro* o *delantero*. Algunas acciones concretas pueden ser *buscar la pelota*, *llevar la pelota*, *pasar la pelota*, *patear al arco*, *esperar un pase*, *despejar la zona*, *cubrir la zona*, *cubrir el arco* o *marcar un oponente*.

Las maniobras son finalmente las actividades más simples. Pertenecen al tipo *Maneuver* y al recibir el mensaje *step*, enviado por la acción que componen, definen el movimiento básico que el robot realizará. Este movimiento básico es del tipo *Movement* y es el encargado de definir las velocidades de las ruedas del robot para realizar su paso.

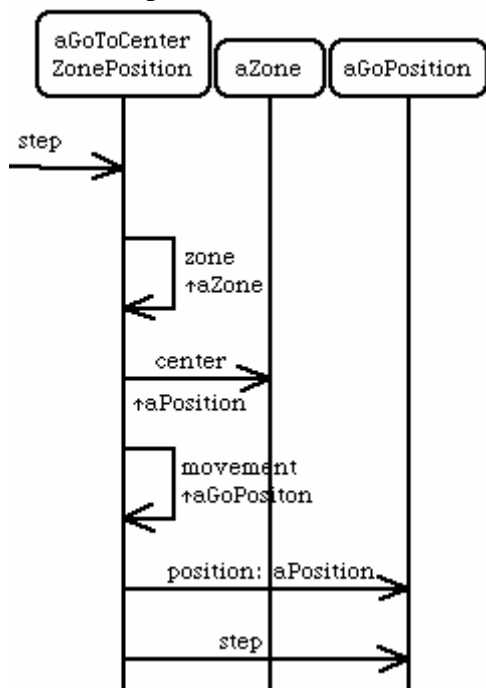


Figura 7. Diagrama de Secuencias. El paso de una maniobra *ir al centro de una zona*.

En el diagrama de la figura 7 puede observarse como la maniobra *ir al centro de la zona* interviene en el paso del robot. Esta puede ser una posible para la acción *cubrir la zona*. El movimiento concreto con que colabora es el más sencillo para un robot de dos ruedas y es *ir a una posición*. En condiciones diferentes, otras posibles maniobras para la acción *cubrir la zona* podrían ser *ir a la posición más*

cercana a la pelota dentro de la zona o *ir a la posición más cercana a un oponente dentro de la zona*.

Movimientos concretos para robots con dos ruedas laterales independientes pueden ser *ir a una posición*, *ir a una posición llegando con una determinada orientación*, *golpear en una posición*, *golpear en una posición con una orientación*, *girar en el lugar* o *permanecer estanco*.

Se proponen las clases abstractas de la figura 8 para definir los tipos *Maneuver* y *Movement*.

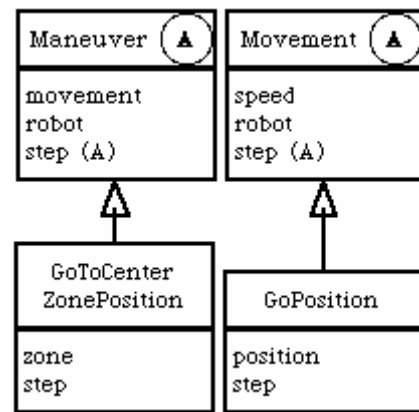


Figura 8. Diagrama de las clases abstractas *Maneuver* y *Movement* con dos subclases posibles. El método asociado al mensaje *step* se define en las subclases.

En el diagrama de instancias de la figura 9 puede observarse las relaciones de conocimiento entre el robot, su actividad y las subactividades sucesivas que intervienen para que el robot realice su paso.

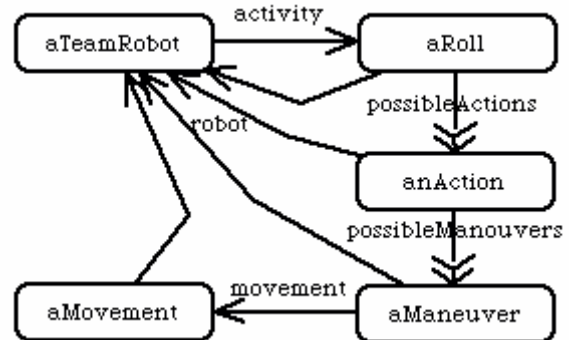


Figura 9. Diagrama de Instancias. Las actividades y subactividades de un robot del equipo.

IV. La elección de subactividades

Como fue descrito, las actividades del tipo *Role* y *Action* delegan, en cada paso, la elección de la subactividad que debe tomar lugar. Lo hacen colaborando con un objeto del tipo *SubactivityElector* que tiene de alguna forma definido el criterio de esta elección. Se propone una clase abstracta homónima para la definición de este tipo (ver figura 10).

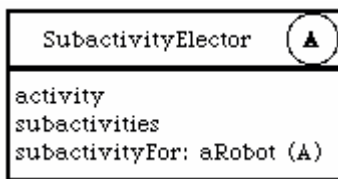


Figura 10. Diagrama de la clase abstracta *SubactivityElector*.

Un criterio sencillo de elección es el de mapeo de estados con subactividades a realizar. De ser éste el criterio deseado, el elector debe contar con una colección de asociaciones entre estados y posibles subactividades. Se puede entender por estado al valor de verdad de un conjunto de premisas evaluadas sobre lo que se sabe del mundo.

Por ejemplo, para la acción *cubrir la zona*, podríamos considerar las premisas *oponente cerca de la zona* y *pelota cerca de la zona*. Tendríamos cuatro valuaciones posibles para este conjunto de premisas y por lo tanto cuatro estados posibles. Al estado que corresponde a la valuación que tiene ambas premisas como falsas –un estado de poco riesgo– podríamos asociarlo a la maniobra *ir al centro de la zona*. Al estado donde sólo la premisa *oponente cerca de la zona* es verdadera podríamos asociarle la maniobra *ir a la posición más cercana a un oponente dentro de la zona*. Los dos estados restantes, que tienen como verdadera la premisa *pelota cerca de la zona*, podríamos asociarlos a la maniobra *ir a la posición más cercana a la pelota dentro de la zona*.

Se sugiere modelar estas premisas con una clase de nombre *Premise* diagramada en la figura 9. La actividad y el robot son el contexto en que se evalúa la premisa.

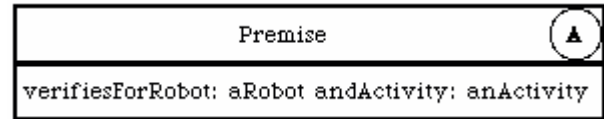


Figura 11. Diagrama de la clase abstracta *Premise*.

También podría incluirse en este diseño un criterio de elección dinámico, que optimice una política de mapeo de estados con subactividades en base a la experiencia de juego.

V. Caso de Estudio

Se describen detalles de una implementación en el lenguaje smalltalk del diseño presentado, probado en el simulador utilizado actualmente en el CAFR.

A cada uno de los cinco robots del equipo se le asignó un rol de los siguientes: *arquero*, *defensor por derecha*, *defensor por izquierda*, *centro* y *delantero*.

La única posible acción para el rol de *arquero* fue *cubrir el arco*. A los roles de *defensor* y *centro* se les definió una zona y asignó como posibles acciones *cubrir la zona*, *despejar la zona*, *buscar la pelota*, *llevar la pelota hacia el arco contrario* y *pasar la pelota*. Al rol *delantero* se le asignó *buscar la pelota*, *llevar la pelota hacia el arco contrario*, *patear la pelota al arco contrario* y *esperar un pase*.

El criterio de elección para todos los roles a excepción del *arquero* quedó definido utilizando aprendizaje por refuerzo [Sutton], en particular Q-Learning [Watkins]. Para ello se definieron estados como valuaciones de premisas y se estableció que toda acción sepa responder si sirve para un determinado estado inicial y si es exitosa según un determinado estado final. Con ello se construyó una función

que para la tupla *<estado inicial, acción, estado final>* retorna un refuerzo de 1 si la acción servía para el estado inicial y había sido exitosa según el estado final, una penalización de 0 si la acción servía para el estado inicial pero no había sido exitosa según el estado final, y una penalización de -1000 si la acción no servía para el estado inicial.

En lo que respecta a las acciones se utilizó siempre el mapeo de estados con maniobras como criterio de elección.

A la acción *cubrir el arco* se le asignaron dos maniobras posibles: *ir a la posición de la línea del arco más cercana a la pelota con orientación paralela a la línea y golpear en la posición de la pelota con orientación opuesta al arco propio*. La primera maniobra quedó asociada a un estado donde la premisa *pelota en poder comfortable* se verifica como falsa y la segunda para cuando se verifica como verdadera.

A las acciones *llevar la pelota hacia el arco contrario* y *patear la pelota al arco contrario* se les asignó como única maniobra posible *golpear en la posición de la pelota con orientación hacia el arco contrario*. A la acción *buscar la pelota* se le asignó como única maniobra *ir a la posición de la pelota*. La acción *cubrir la zona* quedó definida como se describió en el punto III de este documento. Los movimientos implementados para que las maniobras definidas puedan realizarse fueron *ir a una posición*, *ir a una posición con una orientación*, *golpear en una posición* y *golpear en una posición con una orientación*. Se utilizaron para ellos las funciones propuestas por Michael Bowling, Manuela Veloso [Veloso].

Para que esta implementación pueda controlar los robots, se desarrolló una dll en c++ que funcionó de puente entre el ambiente smalltalk y el simulador. Fue necesario contar con un objeto denominado Monitor del Juego. En

cada ciclo del simulador el Monitor del Juego fue encargado de pedir a la dll los datos de posición y orientación de los actores. Luego enviar el mensaje *step* a cada miembro del equipo. Finalmente enviar a través de la dll las nuevas velocidades de los robots.

VI. Trabajos Futuros

Un avance importante será contemplar cambios tácticos o relevos entre miembros del equipo (en el diseño detallado serían cambios de rol). También incorporar más información concerniente al equipo, como su desempeño, u otro dato referente al estado del juego o del ambiente (en el diseño responsabilidades del objeto del tipo *Team*).

Otra mejora quizás sea que los colaboradores posición, orientación y velocidad de los actores estén englobados detrás de uno solo que represente el estado físico del robot (podrían incluirse aquí otras propiedades como velocidad angular, incluso forma del robot, e información acerca de la fidelidad, probabilidad de correctitud o grado de estimación de la información contenida).

También es de suma importancia incorporar mecanismos de predicción, para determinar por ejemplo a qué posición debe dirigirse un robot cuando desea ir al buscar la pelota.

VI. Conclusiones

Se presentó un primer estudio de una porción del dominio del fútbol de robots objetiva. Se distinguieron los actores del juego, se definió una clase para cada uno y otra para el concepto de equipo. Se definió el mensaje *step* para indicarle al robot que debe actuar.

Algunos conceptos no se han detallado en profundidad (rueda, posición, orientación, velocidad, área del campo de juego). Se aconseja modelar cada uno con una clase que refleje lo que estos objetos son, evitando el uso directo de tipos primitivos o predefinidos

como es el caso de los numéricos.

Se presento un segundo estudio correspondiente a una porción subjetiva del dominio del fútbol de robots. Se intentó reducir la diferencia entre la organización de los robots y la organización de las personas a la hora de definir las actividades de los miembros del equipo. Se obtuvo el concepto de rol como actividad primera de los robots. Se definió que sea el encargado de elegir qué acción realizar para cada paso del robot. Se obtuvo el concepto de acción, como subactividad del rol. Se la definió como encargada de elegir la maniobra a realizar para cada paso del robot. Se obtuvo la maniobra como actividad última de todo robot, encargada de definir las características del movimiento del robot.

Luego se desarrolló en particular la forma de incorporar en el diseño mecanismos de elección de subactividades.

Finalmente se comentaron detalles de un caso puntual de implementación del diseño propuesto.

Referencias

[Velo] Motion Control in Dynamic Multi-Robot Environments. [Michael Bowling, Manuela Veloso, Computer Science Department, Carnegie Mellon University]

[Watkins] Q-learning. Machine Learning. [Watkins, C.J.C.H. and Dayan, P. 1992.]

[Sutton] Reinforcement Learning: An Introduction. [Richard Sutton and Andrew Barto.]