

Proyecto Forrest  
Liga de Simulación 2D  
RoboCup

Especificación de Requerimientos

Versión 2.0

Histórico de Revisiones

Versión	Fecha	Resumen de cambios	Autor
1.0	01/12/2005	Versión Inicial	Grupo <i>Forrest</i>
2.0	05/01/2006	Se agrega sección de principales problemas de la liga	Grupo <i>Forrest</i>
3.0	16/10/2006	Revisión general de formato y varios.	Grupo <i>Forrest</i>

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Descripción del Problema</b>	<b>3</b>
<b>3. Requerimientos Funcionales</b>	<b>3</b>
3.1. Comunicación con el Soccer Server . . . . .	3
3.2. Modelado de la realidad . . . . .	3
3.3. Sincronización . . . . .	4
3.4. Estrategia del equipo . . . . .	4
3.5. Habilidades de los jugadores . . . . .	5
3.6. Coach . . . . .	5
<b>4. Requerimientos No Funcionales</b>	<b>5</b>
4.1. Reglas de la Liga . . . . .	5
4.2. Procesos del sistema . . . . .	5
4.3. Plataforma de Desarrollo . . . . .	6
4.3.1. Sistema Operativo . . . . .	6
4.3.2. Lenguaje de Programación . . . . .	6
4.3.3. Sistemas Existentes . . . . .	6
4.3.4. Protocolo de comunicación . . . . .	6
4.3.5. Características de los equipos informáticos a utilizar . . . . .	6
4.4. Performance . . . . .	6
4.4.1. Tiempo de respuesta . . . . .	6
4.5. Confiabilidad . . . . .	6
4.5.1. Disponibilidad . . . . .	6

## 1. Introducción

Este documento tiene como propósito describir los requerimientos funcionales y no funcionales del sistema *Forrest*. Además se presenta un estudio sobre los principales problemas a resolver en el desarrollo de un equipo de fútbol de la liga de simulación 2D de RoboCup.

## 2. Descripción del Problema

El sistema *Forrest* consiste en un equipo de fútbol capaz de competir en la liga simulada 2D de RoboCup [LSIM]. Para la participación en esta liga se requiere interactuar con el simulador de juego *Soccer Server* [CANa], el cual tiene como principal característica ser un sistema de tiempo real. El equipo cuenta con 11 jugadores y un entrenador (Coach). Este último tiene como objetivo observar el partido y si corresponde, tomar acciones que repercutan en la forma de juego del equipo.

Previo al desarrollo de este sistema dadas las características del mismo, se requirió hacer un extenso estudio acerca del estado del arte de la federación, de la liga y de sus principales equipos. El resultado de este trabajo [CANb] sirvió como principal entrada en la elaboración de los requerimientos del sistema *Forrest*, los cuales son presentados en las siguientes secciones.

## 3. Requerimientos Funcionales

Se desea implementar una estrategia para el agente *Forrest*. Para esto se requiere que cada agente, en cada ciclo de simulación, le comunique al simulador el comando que ejecutará. La motivación de este proyecto es incursionar en el problema de la cooperación y planificación en sistemas multi-agente utilizando las técnicas *Grafos de Coordinación y Planeación en Línea de 2 Niveles* presentadas en [KOK] y [MAR].

Para poder construir un equipo completo de la liga simulada de RoboCup es necesario resolver los siguientes problemas.

### 3.1. Comunicación con el Soccer Server

Para que el equipo pueda participar de esta liga debe como primera instancia resolver la comunicación con el simulador de juego *Soccer Server*. Esta comunicación se realiza en ambos sentidos y debe ser resuelta tanto a nivel tecnológico (protocolo de comunicación) como a nivel semántico (protocolo de comandos del simulador).

### 3.2. Modelado de la realidad

Dado que el entorno donde se desempeña el equipo *Forrest* es dinámico y no determinista, se requiere utilizar algún modelo interno de representación de la realidad, que permita al jugador guardar información acerca de como éste percibe el estado actual de su entorno. El principal cometido del modelado de la realidad es proveerle al agente métodos de alto nivel para decidir sus acciones de la mejor forma posible.

Cada jugador requiere conocer su posición y velocidad como así también la de los restantes objetos del campo de juego. En el caso de la liga de simulación, esta información, es provista por el simulador de juego a través del envío de mensajes con información sensorial (física, auditiva y visual).

Un aspecto importante a destacar sobre la información visual es que la misma es parcial e inexacta. Cada jugador solo recibe una porción del estado del juego según su ubicación dentro del campo de juego. A su vez, la información recibida siempre contiene ruido, imposibilitando saber con exactitud la ubicación de un objeto.

Por otro lado, la ubicación de los objetos viene expresada en coordenadas relativas a la posición del jugador que recibe la información. Es por esto que se requiere transformar las coordenadas relativas a coordenadas globales al campo de juego. Esta tarea es inevitable ya que si el agente guardase información con coordenadas relativas a su posición, las mismas quedarían obsoletas ante el primer movimiento del agente luego de haberlas almacenado [CANb].

### **3.3. Sincronización**

El *Soccer Server* puede ser considerado como un sistema de tiempo real que trabaja con intervalos de tiempo discretos usualmente de 100ms de duración denominados ciclos.

En cada ciclo el simulador debe ejecutar los pedidos de acciones de cada jugador y actualizar adecuadamente el estado del juego. Para que la acción de un jugador sea ejecutada en un determinado ciclo  $t$ , la misma debe llegar al simulador durante el transcurso del ciclo  $(t - 1)$ . En el caso de que una acción llegue al servidor en un ciclo distinto al que originalmente fue pensado, la misma puede llegar a ser descartada o ejecutada sobre un estado del juego diferente al planeado. En este caso el resultado de aplicar dicha acción será impredecible. Es por esto que un buen método de sincronización en el envío de acciones al servidor tiene un alto impacto en la performance de los agentes y del equipo.

Un problema importante que dificulta la sincronización es que el jugador desconoce el comienzo de un ciclo, solo sabe su duración.

En resumen, el problema de la sincronización trata de determinar, para cada ciclo, el momento óptimo para enviar una acción al simulador [CANb].

### **3.4. Estrategia del equipo**

La estrategia de un equipo de fútbol en la liga de simulación puede definirse como el plan global y colectivo que siguen sus jugadores durante el partido con el objetivo de ganarle a su adversario. Algunos de los principales aspectos que se deben tener en cuenta en la estrategia son: formación del equipo, roles de cada jugador, forma de comunicación entre los jugadores, jugadas preparadas, comportamientos individuales de cada jugador, dinamismo de la estrategia, etc. [CANb].

Uno de los principales objetivos de la liga de simulación es investigar técnicas de IA y de cooperación en sistemas multi-agente. Hoy en día los equipos participantes tienen una gran madurez en este sentido. Esto hace que sea de sumo interés que el equipo *Forrest* investigue y utilice algunas de estas técnicas en el desarrollo de su estrategia.

### 3.5. Habilidades de los jugadores

Los jugadores deben contar con un conjunto de habilidades básicas que sirvan de soporte para la estrategia del equipo. Algunas de estas son:

- Patear al arco
- Pasar la Pelota
- Posicionarse estratégicamente en el campo de juego
- Avanzar con pelota dominada
- Marcar a un contrario
- Despejar la pelota
- Interceptar la pelota
- Patear penales

### 3.6. Coach

El *Coach* es un agente especial que tiene la posibilidad de visualizar en todo momento la totalidad del campo de juego. Su principal objetivo es dar asistencia a sus jugadores en busca de mejorar la forma de juego del equipo. Para esto debe analizar el comportamiento de los equipos y tomar decisiones que impliquen un cambio en su estrategia. Pudiéndose basar en algunos de los siguientes elementos[CANb]:

- Resultado parcial del partido
- Tiempo de juego restante
- Estrategia del equipo contrario
- Información estadística del partido

## 4. Requerimientos No Funcionales

### 4.1. Reglas de la Liga

Existe un reglamento que debe ser respetado por cualquier equipo que desee participar en competencias oficiales de la liga [CANc]. Es requerimiento del equipo *Forrest* respetar el mismo.

### 4.2. Procesos del sistema

Una de las restricciones más importantes impuestas por el simulador de juego es referido a los procesos con los que debe contar el sistema *Forrest*. Cada uno de los 11 jugadores del equipo y el *Coach* deben ser procesos del sistema independientes, bajo ningún concepto pueden comunicarse directamente entre ellos. La única forma permitida de comunicación entre estos procesos es a través del simulador de juego<sup>1</sup>.

---

<sup>1</sup>El simulador Soccer Server cuenta con varias restricciones acerca de la comunicación entre los jugadores y el coach de un equipo [CANa][CHE].

### **4.3. Plataforma de Desarrollo**

#### **4.3.1. Sistema Operativo**

El sistema debe ejecutarse bajo cualquier distribución del sistema operativo Linux. En particular, en la competencia oficial de la liga del año 2005, se utilizó la distribución *Gentoo*.

#### **4.3.2. Lenguaje de Programación**

En principio el sistema puede estar desarrollado en cualquier lenguaje de programación que pueda ejecutarse bajo el sistema operativo mencionado en el punto anterior.

#### **4.3.3. Sistemas Existentes**

El sistema *Forrest* debe interactuar con el simulador de juego oficial de la liga 2D *Soccer Server* [CANa][CHE]. La versión a utilizar de este sistema es la 10.x.

#### **4.3.4. Protocolo de comunicación**

La comunicación entre los sistemas *Forrest* y *Soccer Server* se realiza a través del protocolo de comunicación UDP/IP. Además, para el envío y recepción de comandos desde y hacia el simulador se debe utilizar el protocolo de comandos definido en [CANa][CHE].

#### **4.3.5. Características de los equipos informáticos a utilizar**

Bajo las condiciones de una competencia oficial<sup>2</sup>, el sistema *Forrest* contará con 3 PCs para distribuir la carga de sus 12 procesos. Cada uno de estos equipos cuentan con un procesador Xeon 3.4Ghz y 1Gb de memoria.

El simulador de juego se ejecuta en un equipo distinto a estos 3 PCs.

### **4.4. Performance**

#### **4.4.1. Tiempo de respuesta**

El simulador de juego en condiciones de competencia, avanza con intervalos de tiempo discretos de 100ms de duración llamados ciclos. Al termino de cada ciclo el simulador actualiza el estado del juego con las acciones que le fueron enviadas por los jugadores de ambos equipos (una acción enviada al simulador en el ciclo  $x$  es ejecutada en el ciclo  $x+1$ ). Este funcionamiento determina que si un jugador no envía una acción en ciclo  $x$ , no ejecutará acción en el siguiente ciclo  $x+1$ .

Por lo dicho anteriormente, se requiere que el proceso de toma de decisión de una acción para cada jugador no demore más de 100ms.

### **4.5. Confiabilidad**

#### **4.5.1. Disponibilidad**

El sistema debe estar disponible durante el tiempo que demore la ejecución del partido (10 minutos).

---

<sup>2</sup>Datos relevados hasta la competencia de Osaka 2005 [CANc].

## Referencias

- [CANa] Canales, R.; Casella, S.; Rodríguez, P. "*Características del Simulador 2D*". Tejera, G. (tutor). Trabajo de Grado. Facultad de Ingeniería. Instituto de Computación - Montevideo, Junio 2005.
- [CANb] Canales, R.; Casella, S.; Rodríguez, P. "*Estado del Arte, RoboCup - Liga Simulada*". Tejera, G. (tutor). Trabajo de Grado. Facultad de Ingeniería. Instituto de Computación - Montevideo, Setiembre 2005.
- [CANc] Canales, R.; Casella, S.; Rodríguez, P. "*Reglas de la Liga - Reglas de la Competencia Osaka 2005*". Tejera, G. (tutor). Trabajo de Grado. Facultad de Ingeniería. Instituto de Computación - Montevideo, Junio 2005.
- [CHE] Chen, M.; Dorer, K.; Foroughi, E.; Heintz, F.; Huangy, Z.; Kapetanakis, S.; Kostiadis, K.; Kummeneje, J.; Murray, J.; Noda, I.; Obst, O.; Riley, P.; Steens, T., Wang, Y.; Yin, X. "*RoboCup SoccerServer: User Manual*". Comunidad de RoboCup, Febrero 2003.
- [KOK] Kok, J.; Spaan, M.; Vlassis, N. "*Non-communicative multi-robot coordination in dynamic environments*". Robotics and Autonomous Systems. En press, 2005.
- [LSIM] Liga simulada de RoboCup. <<http://sserver.sourceforge.net/index.html>>[Consulta: Agosto 2005].
- [MAR] Martínez Ledesma, J. "*Aplicación de un método de Planificación en Línea en un Equipo de la Liga de Simulación de RoboCup*" PhD tesis. Instituto Tecnológico y de Estudios Superiores de Monterrey, Mexico, 2005.