



**CONACYT-México  
CNRS-INRIA-Universidad Joseph Fourier Grenoble I –Francia**

*Formato-proyecto 2004*

**Laboratorio Franco-Mexicano de Informática, LAFMI**

Nombre del coordinador mexicano del proyecto: **Alfredo Weitzenfeld**

Nombre del coordinador francés del proyecto: **Peter Ford Dominey**

**1. Título y tipo de proyecto:**

Título: **Una Interfaz de Lenguaje Natural para el Comando y Control de Robots**

Palabras clave (máximo 4) : **tecnología de lenguaje humano, robots autónomos**

**Seleccionar una opción:**

Investigación Básica:

Investigación Aplicada:  X

Desarrollo Tecnológico:

**2. Instituciones:**

**En México**

Institución del Coordinador del Proyecto: **Instituto Tecnológico Autónomo de México, ITAM**

Laboratorio (o equipo): **Div. Académica de Ingeniería**

**En Francia**

Institución del Coordinador del Proyecto: **Institut des Sciences Cognitives**

Laboratorio (o equipo) : **Sequential Cognition and Language Group**

Estatus de la unidad : **CNRS UMR 5015**

## 5. Descripción detallada del proyecto

### 5.1 Identificación del tema y estado del arte:

**Tema : Interacción humano-robot en tiempo real para comportamiento dirigido a metas.**

Según los robots se vuelven cada vez más capaces de funciones sensoriales y motrices complejas, con la habilidad de llevar a cabo tareas cada vez más complejas, la habilidad de interactuar con ellos de forma ergonómica, en tiempo real y de manera adaptiva se vuelve una preocupación cada vez más importante. El proyecto actual busca combinar el estado del arte en el control de robots autónomos con el estado del arte de los sistemas cognitivos para la interacción basada en lenguaje para proveer una interfaz humano-robot basada en lenguaje robusta.

Idealmente, la investigación en la Interacción Humano-Robot (IHR) permitirá una comunicación y colaboración natural, ergonómica y óptima, entre los humanos y los sistemas robóticos. Para avanzar en esta dirección, hemos identificado dos requisitos mayores: Primero, debemos estudiar un ambiente robótico real donde los tecnólogos e investigadores hayan desarrollado una amplia experiencia y un conjunto de necesidades con respecto a IHR. Segundo, debemos estudiar un sistema de procesamiento de lenguaje independiente de un dominio particular que tenga validez psicológica, y que pueda ser relacionada a dominios arbitrarios. En respuesta al primer requerimiento relacionado al contexto robótico, estudiaremos el Comando y Control de Robots en el contexto internacional de robots que juegan fútbol, donde el grupo Mexicano compite a nivel internacional. Desde un contexto de lenguaje psicológicamente válido, estudiaremos un modelo de lenguaje y significado correspondiente desarrollado por el laboratorio Francés que ha descrito tanto aspectos neurológicos como de comportamiento en el lenguaje humano, y lo ha implantado en contextos robóticos.

#### Estado del Arte: A. Comando y Control de Robots en el Dominio AIBO de Robocup

**RoboCup** [3] (originalmente llamado Robot World Cup Initiative) es una iniciativa de investigación y educación internacional que provee un dominio de problema unificado de fútbol para la robótica inteligente. En tal ambiente dinámico, múltiples robots de rápido movimiento deben colaborar en lograr su objetivo de meter goles bajo diferentes estrategias de grupo. Existen múltiples ligas en RoboCup cada una con sus propios desafíos como visión global versus local, ruedas versus patas, hardware versus software, simulado versus tiempo real, etc. El común denominador entre estas ligas es la autonomía de los robots.

**Liga AIBO de Sony** [1] es de interés particular para nuestro proyecto ya que es la única liga de robots reales donde no es necesario que el equipo construya su propio hardware. La liga AIBO utiliza una plataforma de software estandarizada, Open R [2], donde los equipos deben desarrollar sus propios algoritmos de software basados en una arquitectura de hardware única. Cada equipo consiste de cuatro robots de cuatro patas, donde cada robot incorpora su propio procesador, articulaciones de patas y cabeza, cámara de video y sensores que le permiten interactuar con el mundo externo. Existe un número de desafíos que deben ser resueltos para que los AIBOs jueguen al fútbol. Primero, el robot debe procesar imágenes en tiempo real para poder identificar objetos en el campo de fútbol (jugadores, porterías, pelota, etc.). Además, el robot debe poder conocer su ubicación en el campo, debe poder caminar con sus cuatro patas y ejecutar acciones especializadas como patear la pelota hacia la portería y bloquear tiros entrantes en el caso de un portero. El robot debe poder tomar decisiones sobre qué acciones (comportamientos) y cuándo deben llevarse a cabo, mientras comparte esta información con los demás robots en el desarrollo de una estrategia de equipo.

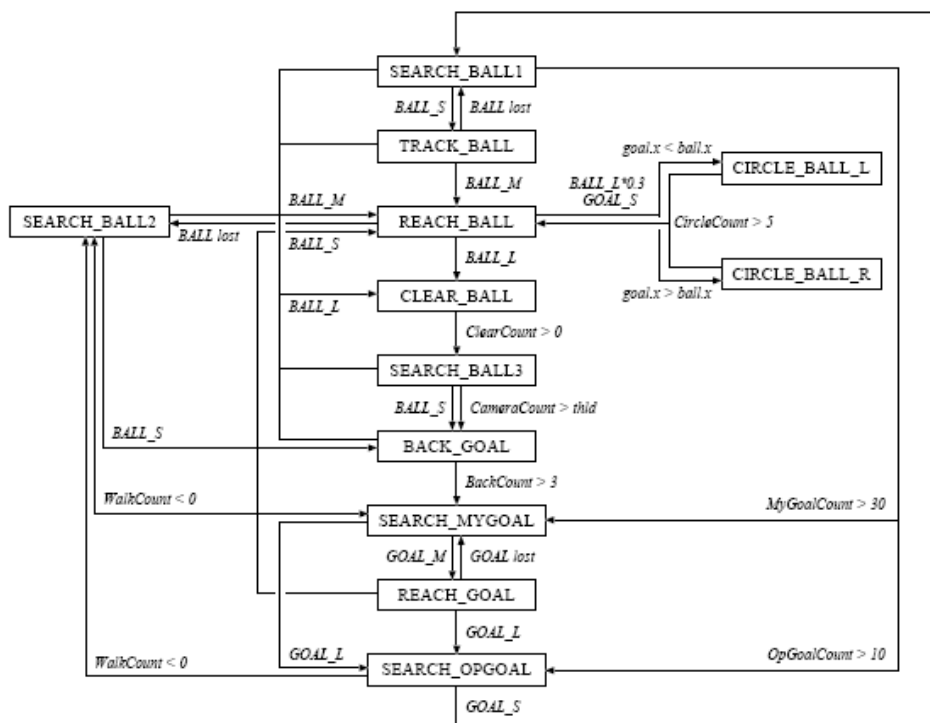
La liga AIBO presenta un número de desafíos interesantes para robots autónomos que colaboran:

- **Percepción.** El reconocimiento de objetos visuales depende tanto de consideraciones internas como externas al robot. En particular, condiciones externas como calibración de iluminación y color afectan de manera crítica el completo desempeño del robot. Por ejemplo, bajo condiciones de iluminación pobre, algunos colores pueden confundirse con otros, causando que el robot perciba información errónea, por lo tanto afectando la localización. Por otra parte, diferentes tonos de colores similares pueden también

confundir a los sistemas de percepción de los robots. A pesar de que los algoritmos de reconocimiento más complejos pueden resolver estos problemas, tal enfoque afectará directamente la eficiencia de procesamiento.

- **Movimiento.** El caminado de robots de cuatro patas requiere movimiento de articulaciones coordinado. Los movimientos rápidos tienen un efecto directo en el rendimiento del juego. Sin embargo, las percepciones y los movimientos deben sincronizarse para obtener resultados óptimos.
- **Coordinación.** El juego en equipo tiene un importante efecto en el resultado del juego. Mientras que los robots deben ser individualmente eficientes, la coordinación entre robots es clave para lograr un rendimiento de juego exitoso contra equipo avanzados. La coordinación de acuerdo a las estrategias varía dependiendo de las habilidades de los robots individuales, número de jugadores en un momento particular en el juego, etc. Es crítico por lo tanto poder experimentar con diferentes estrategias y evaluar su rendimiento contra otros equipos y condiciones de juego.
- **Modelo del mundo.** Aprovechando la comunicación inalámbrica entre robots, es posible construir un modelo compartido único del campo y de la ubicación de la pelota y de los robots. Tal modelo reduce la incertidumbre en la localización del robot y enriquece las estrategias de coordinación entre robots. Al percibir la pelota, un robot lejano puede informar a miembros del equipo mejor posicionados a reorientarse y moverse hacia la pelota.
- **Reconocimiento de acciones.** Existen muchas áreas de mejora en las estrategias de juego actuales de AIBOs. Por ejemplo, la mayoría de los equipos no prestan ninguna atención al reconocimiento de acciones en el campo a diferencia del reconocimiento más tradicional de objetos individuales. En el reconocimiento de acciones, los robots pudieran anticipar acciones por parte del equipo contrario reconociendo situaciones del juego, como pateo de pelota o bloqueo de un miembro del equipo.

Las siguientes son las estrategias de AIBO más comunes usadas por los equipos: (i) comportamiento individual sin colaboración, (ii) colaboración con roles estáticos, y (iii) colaboración con roles dinámicos.



**Figura 1.** Transiciones de estados para un portero.

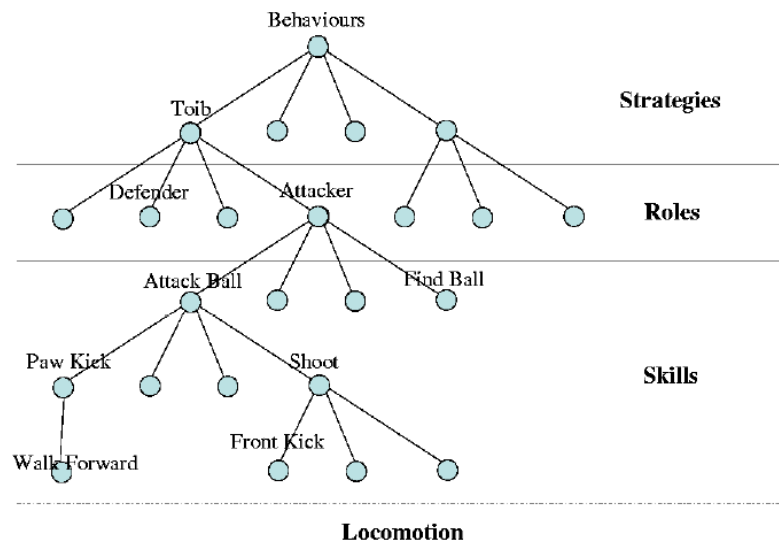
**Comportamiento Individual sin Colaboración:** Esta es la estrategia de comportamiento más sencilla utilizada inicialmente por muchos equipos [6]. La estrategia individual consiste generalmente en tener dos tipos de jugadores: portero y atacantes. En la Figura 1 se muestra un ejemplo de transición de estados para un portero.

La estrategia de un atacante usualmente involucra búsquedas de la pelota dando vueltas en el campo. Cuando encuentra la pelota, el robot se acerca y se para frente a ella verificando su orientación para patear la pelota en la dirección correcta. La auto-localización es importante tanto para una orientación correcta como para evitar la regla de defensa ilegal donde ningún jugador con excepción del portero puede estar presente en el área de la portería.

**Colaboración con Roles Estáticos:** Una estrategia básica de colaboración es asignar roles estáticos a los jugadores. Por ejemplo, en [7] los robots tienen tres roles separados: atacante primario, apoyador ofensivo, y apoyador defensivo. En este modelo, cada robot requiere un agente diferente para negociar con otros agentes las jugadas a ser hechas. Para lograr esta estrategia un modelo global del mundo es usualmente construido, donde los robots comparten información entre ellos. La información compartida incluye la posición actual del robot y un estimado de incertidumbre para esa posición; estimación de la posición de la pelota y la incertidumbre asociada con ella.

Los jugadores de campo tienen su rol asignado de manera fija como: atacante primario, apoyador ofensivo, y apoyador defensivo, respectivamente. El atacante primario se especifica primero, seguido del apoyador defensivo, y finalmente del apoyador ofensivo. Este orden está diseñado para hacer el sistema más robusto. Si uno o dos de los robots fallan, los miembros restantes del equipo continúan jugando.

**Colaboración con Roles Dinámicos:** Una estrategia de colaboración más compleja involucra robots con roles dinámicos, donde un enfoque común es utilizar un árbol de decisión combinando información de la entrada visual y el modelo del mundo existente para definir el siguiente conjunto de acciones a ser ejecutadas. El árbol es generalmente dividido en tres niveles, correspondientes a estrategias, roles y habilidades [5,6], como se muestra en la figura 2. El nivel de más arriba define las estrategias que pueden utilizarse. Por ejemplo, una estrategia pudiera tener a los robots esparcidos en la cancha pasándose entre sí la pelota. Otra estrategia pudiera hacer que los robots jueguen como atacantes, y otra pudiera tener a todos los robots jugando defensivamente. A este nivel las diferentes estrategias pueden desarrollarse y probarse entre sí. Durante un juego, los robots pueden programarse para cambiar estrategias dependiendo del resultado actual, o de cuanto tiempo queda en el partido.



**Figura 2.** Árbol de decisión para la colaboración de robots con roles dinámicos.

Cada estrategia incorpora un número de roles que los robots pueden ejecutar. Ejemplos incluyen, atacantes, apoyadores y defensas. Estas estrategias se cambian generalmente de manera dinámica durante un partido, con excepción del rol de portero asignado de manera fija a un robot particular. En el nivel más bajo cada robot contiene un número de habilidades.

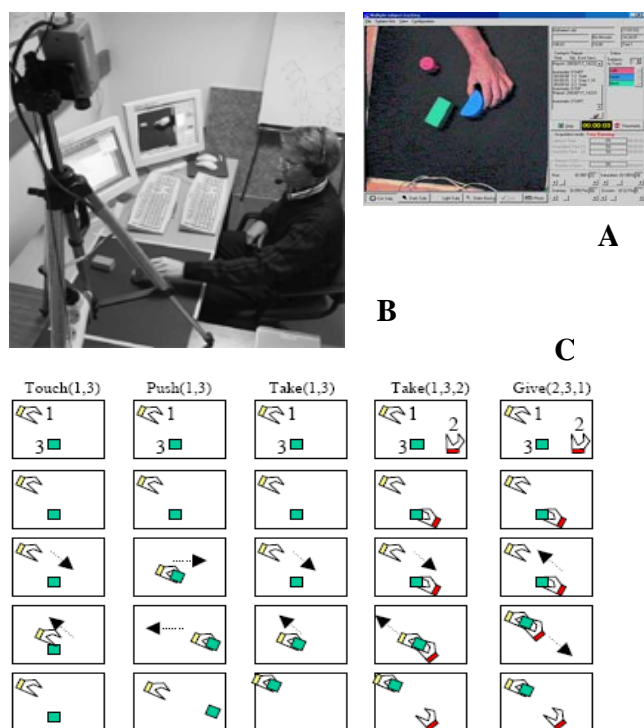
## Estado del Arte: B. Interacción Humano-Robot

Basado primordialmente en desarrollos recientes en tecnología de voz, robótica, y poder de cómputo, la

interacción Humano-Robot se ha vuelto un dominio de investigación de rápido desarrollo e importancia. Esto se refleja en la existencia a partir de 1997 de la Conferencia Internacional en Interfaces de Usuario Inteligentes, y la inclusión de Interacción Humana en la Conferencia Internacional de Robots Humanoides IEEE-RAS/RSJ, iniciada en el 2000. La investigación en este dominio ha revelado que cuatro aspectos críticos deben ser considerados en un contexto interacción humano-robot exitoso. Primero, el ambiente de comportamiento debe estar bien definido, y ser lo suficiente rico para beneficiar tal interacción. El ambiente Aibo de RoboCup descrito anteriormente claramente cumple con estos requisitos. Segundo, el uso de medios más allá de lo lingüístico, incluyendo visión, debe ser claramente caracterizado. Tercero, el sistema debe utilizar un modelo de lenguaje apropiado para lograr un mapa hacia atrás y adelante de oraciones y significado, y finalmente, todo esto debe ser parte integral de un contexto ingenieril de software robusto para construir sistemas tecnológicos de lenguaje humano confiable.

**Procesamiento de Eventos Visuales:** Uno de los problemas principales en visión por computadora ha sido el desarrollo de algoritmos robustos para interpretar escenas y eventos, una vez la identificación y seguimientos de los objetos ha sido lograda. Explotando la idea que las primitivas perceptuales pueden proveer la base para esta robustez, Siskind [15] ha demostrado que las primitivas dinámicas de fuerza de contacto, apoyo, y adhesión pueden extraerse de las secuencias de eventos de video y utilizadas para reconocer eventos incluyendo levantado, puesta, y apilado basados en su caracterización en una lógica de eventos. Resultados relacionados han sido logrados por Steels y Baillie [16]. El uso de estas representaciones intermedias permite robustez del sistema a variabilidad en los parámetros de movimiento y vista. Lo más importante que esta investigación demostró es que la semántica léxica para un número de verbos puede ser establecido mediante un procesamiento automático de imágenes.

Hemos recientemente explotado este enfoque, utilizando la primitiva perceptual de contacto para caracterizar eventos físicos, como se muestra en la Figura 3 [19].



**Figura 3.**

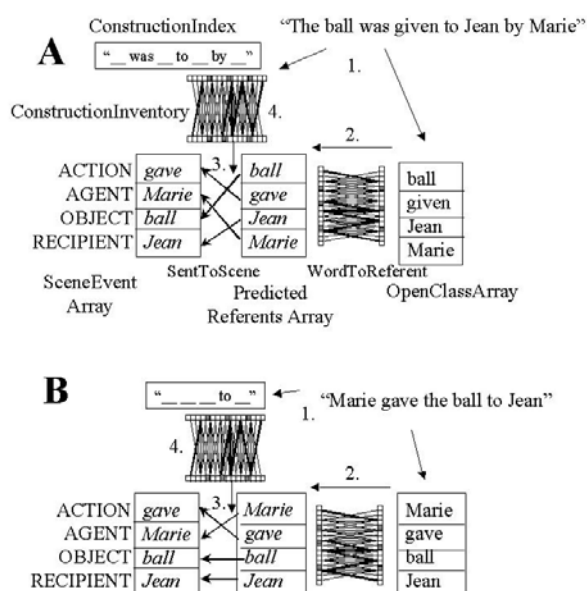
A. Sistema de Interacción Humano-Robot. El operador Humano ejecuta eventos con bloques, se los describe al robot para su aprendizaje, y luego le hace preguntas respecto a los eventos durante la demostración de aprendizaje.

B. Imagen de la entrada visual al robot. C. Plantilla temporal de eventos de contacto que permiten al sistema de visión reconocer los eventos de tocar, empujar, tomar y dar.

**Tecnología de Lenguaje Humano y Modelado de Lenguaje:** Tecnología de Lenguaje Humano (TLH) se refiere a sistemas para el reconocimiento de voz y conversión a texto, síntesis de voz y flujo de diálogo de control. Estos aspectos han sido sujeto de financiamiento extenso por parte de organizaciones como DARPA (Defense Advanced Research Projects Agency), y ha resultado en el desarrollo de herramientas TLH de dominio público incluyendo el *Communicator* de CMU [17], y el *Toolkit* de CSLU [18]. Hemos utilizado el ambiente de Desarrollo Rápido de Aplicaciones de CSLU descrito más adelante.

Un aspecto crucial en las interfaces humano-robot es la manera por la cual el significado es intercambiado mediante lenguaje natural. En este contexto, existen diferentes métodos clásicos incluyendo gramáticas libres de contexto y modelos *n-gram*, y estas gramáticas pueden explotarse directamente en paquetes de software existentes incluyendo el *Toolkit* de CSLU, y *CPK-NLP Suite*. En este contexto, hemos desarrollado un modelo de lenguaje basado en la teoría de gramática de construcción, que tiene la ventaja sobre muchos otros enfoques que puede aprender a acomodar nuevas estructuras gramaticales mediante entrenamiento. Hemos demostrado la utilidad del enfoque de gramática de construcción en una plataforma de interacción humano-robot.

La Figura 3 ilustra la configuración de la plataforma de interacción humano-robot. Esta plataforma fue inicialmente utilizada para generar pares <Oración, Significado> para el entrenamiento del modelo de construcción gramatical ilustrada en la Figure 4. Las oraciones son extractos de voz hablada usando convertidores comerciales de voz a texto, y los significados son extraídos de los eventos de las imágenes de video mediante un *parsing* de secuencias temporales de contactos, descritas en la Figura 3C. Una vez se entrenó el modelo, las construcciones gramaticales aprendidas sólo pueden ser usadas por el modelo para generar la oración correspondiente para el significado dado. Esto ha sido explotado en un sistema de descripción de escenas y preguntas-respuestas [19].



**Figure 4.** Arquitectura de construcción gramatical. Procesamiento de tipos de oración activos y pasivos en A y B, respectivamente. A partir de una entrada, las palabras de *OpenClass* rellenan el arreglo *OpenClassArray* (OCA), y las palabras de *ClosedClass* rellenan *ConstructionIndex*. Los elementos del *VisualSceneAnalysis* rellenan a *SceneEventArray* (SEA) con significado extraído de los elementos de la escena. Las palabras en OCA son traducidas a *PredictedReferents* mediante el mapeo de *WordToReferent* para rellena *PredictedReferentsArray* (PRA). Elementos de PRA son mapeados a sus roles en *SceneEventArray* (SEA) mediante mapeo de *SentenceToScene*, específico a cada tipo de oración. El mapeo es obtenido de *ConstructionInventory*, mediante *ConstructionIndex* que codifica las palabras de *ClosedClass* que caracterizan cada tipo de oración.

## 5.2 Objetivos específicos:

Dado estos antecedentes, el objetivo principal de este proyecto es desarrollar una interfaz humano-robot robusta y confiable basada en lenguaje natural. Esta interfaz va a ser entre Robots Aibos en el contexto RoboCup, y *coaches* que dirigen la actividad de los robots. El ambiente RoboCup ha sido escogido porque es un ambiente robótico estandarizado y bien documentado en el cual el ITAM tiene amplia experiencia, y por lo tanto provee un dominio cualitativo para la evaluación del éxito. Este objetivo es consistente con los ejes mayores de la llamada a propuestas con respecto a agentes basados en sistemas distribuidos, procesamiento de imágenes, robótica y comunicación.

Mientras que la comunicación debiera ser relativamente gratuita, ésta será organizada alrededor de la estructura de la tarea, en otras palabras el dominio RoboCup. Inclusive, un sub-objetivo es demostrar que la representación *predicado(argumento)* del significado de las frases en lenguaje natural pueden generalizarse a diferentes dominios, incluyendo la tarea de Robocup. Por lo tanto, en esta tarea, las interacciones humano – robot pueden dividirse en cuatro categorías:

1. **Comandos de juego:** Instrucciones específicas sobre qué hacer – incluyendo *Patear*, *Pasar la pelota a X*, *Defender la portería*, etc.

2. **Interrogación de estado:** Preguntas incluyendo - que está haciendo, donde está, que tan lejos está de la pelota, etc.
3. **Interrogación de justificaciones:** Principalmente concerniente con determinar por qué el robot ejecutó cierta acción.
4. **Estrategia del entrenador** (“coach”): Transmisión de conocimiento de estrategia, por ejemplo. “Si se está bloqueado en el frente, pase la pelota a uno de los jugadores detrás suyo.”

Para cada una de estas interacciones estableceremos una relación entre lenguaje y significado (en el contexto de control de robots), con estas relaciones representadas como construcciones comunicativas que pueden utilizarse de manera generalizada. Por lo tanto, los comandos de juego involucrarán construcciones comunicativas que transforman comandos como “Pasa la pelota a Juan” a sintaxis de comandos de robot con la forma “*pasa(pelota, Juan)*”. Interrogación de estado e interrogación de justificación que son transformadas en consultas que interrogan las estructuras de datos de control del sistema. Las respuestas a estas consultas serán entonces transformadas a oraciones comprensibles humanas que serán tocadas por un sintetizador de voz. Las interacciones de estrategia del entrenador modificarán las reglas de la base de datos que definen comportamiento como “*si posee(pelota) y portería(bloqueada) entonces pasa(pelota)*”.

## 5.3 Metodología:

### 5.3.1 Definir construcciones comunicativas para diferentes tipos de interacción

Esto involucrará para cada tipo de comando, un mapeo de la estructura gramatical del comando a una estructura *predicado(argumento)* para el comando a nivel de Aibo.

Ejemplos de instrucciones del entrenador a los atacantes:

- a. A un atacante:
  1. *Patear*. Cuando un jugador tiene la pelota, el entrenador puede ordenar al jugador que patee la pelota. Esta acción puede ser usada para patear la pelota en dirección de la portería del equipo contrario o para patearla lejos de su propia portería.
  2. *Pasar la pelota*. Cuando un atacante diferente a otro cercano a la pelota tiene una mejor posición para tomar un tiro, el entrenador puede ordenar al atacante cercano a la pelota que le pase la pelota al otro atacante.
  3. *Defender un tiro libre*. Actualmente, el partido no se para a causa de un tiro libre, sin embargo esta regla puede cambiar en el futuro. En tal caso, el entrenador puede ordenar al robot ir a defender el tiro libre para evitar un tiro directo sobre la portería por parte de un jugador contrario.
- b. A múltiples atacantes:
  1. *Atacantes defender*. Cuando un atacante pierde la pelota el equipo puede volverse más vulnerable a un contraataque del equipo contrario. El entrenador puede ordenar a los atacantes que regresen a la portería y la defiendan.

Ejemplos de instrucciones del entrenador al portero:

1. *Portero avanzar*. En algunas ocasiones el portero no saldría a atrapar una pelota ya que la pelota puede que esté fuera de alcance. Existen algunas situaciones cuando lo opuesto es deseado, por ejemplo, para evitar un tiro del atacante contrario. El entrenador puede ordenar al portero que salga y atrape la pelota.

Ejemplos de instrucciones del entrenador a los defensas:

1. *Retener la pelota*. Existen algunas ocasiones cuando quisiéramos que un jugador retenga la pelota. Esta acción puede ser usada cuando otros jugadores son retirados del campo. El entrenador puede ordenar a un defensor a que retenga la pelota.
2. *Pasar la pelota*. Similar a *pasar la pelota* para un atacante.

Ejemplos de instrucciones del entrenador a cualquier jugador:

1. *Parar.* Parar todas las acciones para evitar una falta u obstruir un tiro de su propio equipo.
2. *Localizar.* Cuando el entrenador ve que un jugador está perdido en el campo, le puede ordenar al jugador que se localice nuevamente en el campo.

Ejemplos de instrucciones del entrenador a todos los jugadores:

1. *Defender.* Defender con todos los jugadores. Todos los jugadores se mueven a una posición defensiva.
2. *Atacar.* Atacar con todos los jugadores (excepto portero). Todos los jugadores se mueven a una posición atacante.

Ejemplos de consulta del entrenador a cualquier jugador:

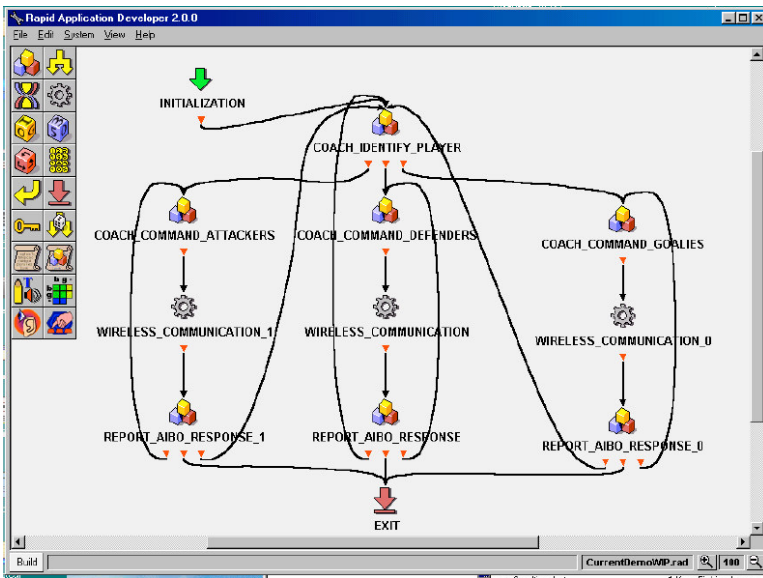
1. *Su acción.* El jugador devuelve la acción que está tomando actualmente.
2. *Su localización.* El jugador devuelve su localización en el campo.
3. *Su distancia a la pelota.* El jugador devuelve su distancia a la pelota.
4. *Objetos que puede ver.* El jugador devuelve todos los objetos que ve (marcas, jugadores, portería y pelota).
5. *¿Por qué has tomado esa acción?* El jugador devuelve las razones para tomar una acción particular. (Por ejemplo, el jugador estaba cerca a la pelota y vio la portería, por lo cual el jugador pateó la pelota a la portería.)
6. *Su comportamiento actual.* El jugador devuelve su comportamiento actual (atacando, defendiendo, etc.)

Para cada uno de los tipos de interacción descritos anteriormente, definiremos la construcción comunicativa que identifica el mapeo estructural entre oraciones gramaticales y comandos en el protocolo de interacción del robot.

### 5.3.2 Definir el modelo de diálogo y protocolo de interacción

Una vez se define la interfaz de nivel de comando, debe ésta insertarse en la estructura general de intercambio de “ida y vuelta” del diálogo Humano-Robot. Concretamente, abajo se muestran dos ejemplos de diálogo entre entrenador y atacante, y entre entrenador y portero. En la figura 5 se muestra un diagrama del modelo de diálogo prototipo para las interacciones Entrenador-Aibo.

Ejemplo 1. Entrenador instruyendo a un atacante.	Ejemplo 2. Entrenador instruyendo al portero.
<p><b>Coach:</b> ¿Ve la pelota?  <b>AIBO:</b> No, no la veo.  <b>Coach:</b> La pelota está detrás suyo. Gira 180 grados.  <b>AIBO:</b> Ok.  <b>Coach:</b> ¿Qué objetos ve?  <b>AIBO:</b> Sólo veo la pelota.  <b>Coach:</b> ¿Cuál es su distancia a la pelota?  <b>AIBO:</b> 30 centímetros.  <b>Coach:</b> Vaya hacia la pelota  <b>AIBO:</b> Ok.  <b>Coach:</b> Ahora pasa la pelota al AIBO 2.  <b>AIBO:</b> ¿Cuál es la posición del AIBO 2?  <b>Coach:</b> La posición del AIBO 2 es x, y.  <b>AIBO:</b> Ok.  <b>Coach:</b> ¿Cuál es su acción actual?  <b>AIBO:</b> Estoy girando a la derecha 40 grados.  <b>AIBO:</b> Ahora estoy pasando la pelota al AIBO 2.  <b>Coach:</b> Ok. Ahora regrese a su portería.  <b>AIBO:</b> Ok.</p>	<p><b>Coach:</b> ¿Ve la pelota?  <b>AIBO:</b> Sí.  <b>Coach:</b> ¿Cuál es su distancia a la pelota?  <b>AIBO:</b> Más de 60 centímetros.  <b>Coach:</b> Ten cuidado. El equipo opuesto tiene la pelota.  <b>AIBO:</b> Ok.  <b>Coach:</b> Si ve la pelota a una distancia menor de 40 centímetros, avanza a tomar la pelota.  <b>AIBO:</b> Ok.  <b>Coach:</b> ¿Cuál es su acción actual?  <b>AIBO:</b> Estoy yendo a tomar la pelota.  <b>Coach:</b> ¿Por qué has tomado esa acción?  <b>AIBO:</b> Ví la pelota a 30 centímetros de mi posición, por lo cual seguí su instrucción.  <b>Coach:</b> Ok.</p>

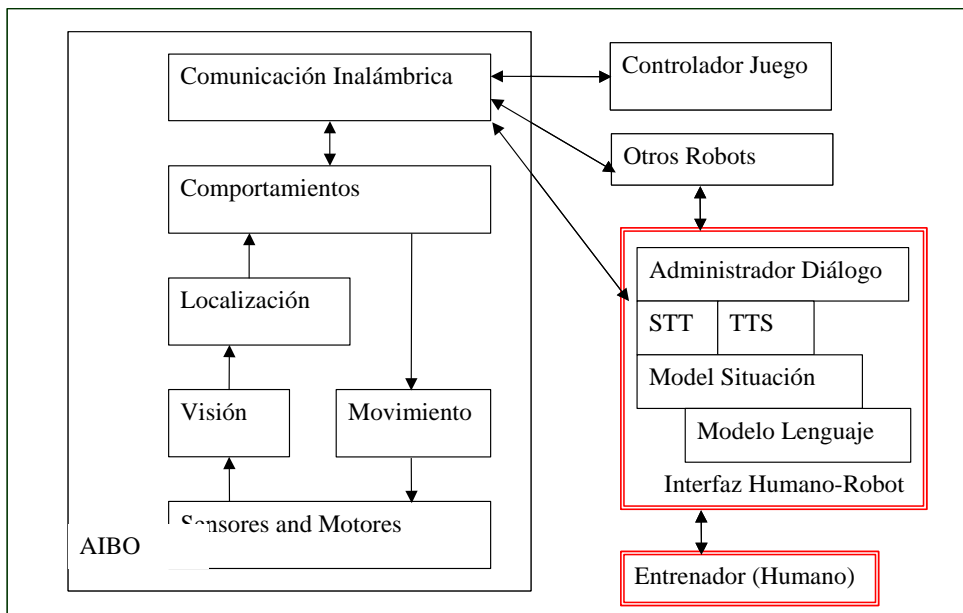


**Figura 5.** Modelo de diálogo prototipo para las interacciones Entrenador-Aibo. Para poder restringir el reconocimiento de voz, el entrenador primero identificará al jugador(es) con quién está hablando, y luego entrará en una interacción de COMANDO específica al dominio. Los comandos serán transmitidos al AIBO mediante comunicación inalámbrica, el Aibo ejecutará/responderá y cualquier mensaje será dicho al entrenador mediante un sintetizador. El entrenador puede entonces escoger continuar la interacción con los mismos jugadores (como en el Ejemplo 1), o cambiar a un nuevo jugador (como en la transición del Ejemplo 1 al Ejemplo 2).

### 5.3.3 Asignar estas funciones a software y hardware

**Arquitectura del Sistema.** El sistema de jugar fútbol del AIBO incluye algoritmos de percepción y control especializados con liga al sistema operativo de Open R. Open R ofrece un conjunto de interfaces modulares para acceder a diferentes componentes de hardware en el AIBO. Los equipos son responsables de la programación a nivel aplicación, incluyendo el diseño de la arquitectura de control de percepción y movimiento del sistema. En la Figura 6 se muestra una arquitectura de sistema típica del AIBO. La arquitectura incluye los siguientes módulos:

- **Sensores and Motores.** Este módulo interactúa con las llamadas de bajo nivel de OPEN-R para controlar los motores físicos y obtener información de la cámara.
- **Visión.** Recibe una secuencia de imágenes en tiempo real de la cámara del AIBO. Ejecuta el procesamiento necesario para identificar los objetos en el campo. El nivel más sencillo de reconocimiento involucra identificación de objetos por color con filtrado adicional para evitar inconsistencias y optimizar el ciclo de procesamiento visual.
- **Localización.** Determina la posición del robot tomando en cuenta la portería, borde del campo y marcadores en las esquinas del campo. Diferentes algoritmos son usados para incrementar el grado de confianza con respecto a la posición de cada robot. Los robots comparten esta información para obtener un modelo del mundo.
- **Comunicación Inalámbrica.** Transfiere información entre robots en el desarrollo de un modelo global o una estrategia coordinada. Recibe información del Controlador de Juego, una computadora remota enviando información sobre el estado del juego (gol, falta, inicio y fin del juego) controlado por un árbitro humano. Provee la base para la Interacción Humano-Robot.
- **Comportamientos.** Controla movimientos del robot a partir de los comportamientos programados en respuesta a información de otros módulos, como visión, localización y comunicación inalámbrica. Los comportamientos son afectados por estrategia de juego, roles específicos que los jugadores toman, tales como atacante o portero, y mediante interacción humana.
- **Movimiento.** Recibe comandos de los módulos de comportamiento correspondientes a las acciones de los robots, tales como caminar, correr, patear la pelota, girar a la derecha o izquierda, etc. Estas acciones controlan los motores en las patas y la cabeza.



**Figura 6.** Arquitectura de sistema del robot AIBO. Los módulos son desarrollados por cada equipo con acceso al hardware mediante llamadas al sistema Open R. Los subsistemas Entrenador e Interfaz Humano-Robot corresponden a los nuevos componentes para la interacción humano-robot. Esto incluye el Administrador de Diálogo (implementado en CSLU RAD), el “Speech To Text” (STT) y “Text To Speech” (TTS), el “Modelo Situación”, y el “Modelo Lenguaje”.

#### 5.3.4 Definir acuerdos de interfaz

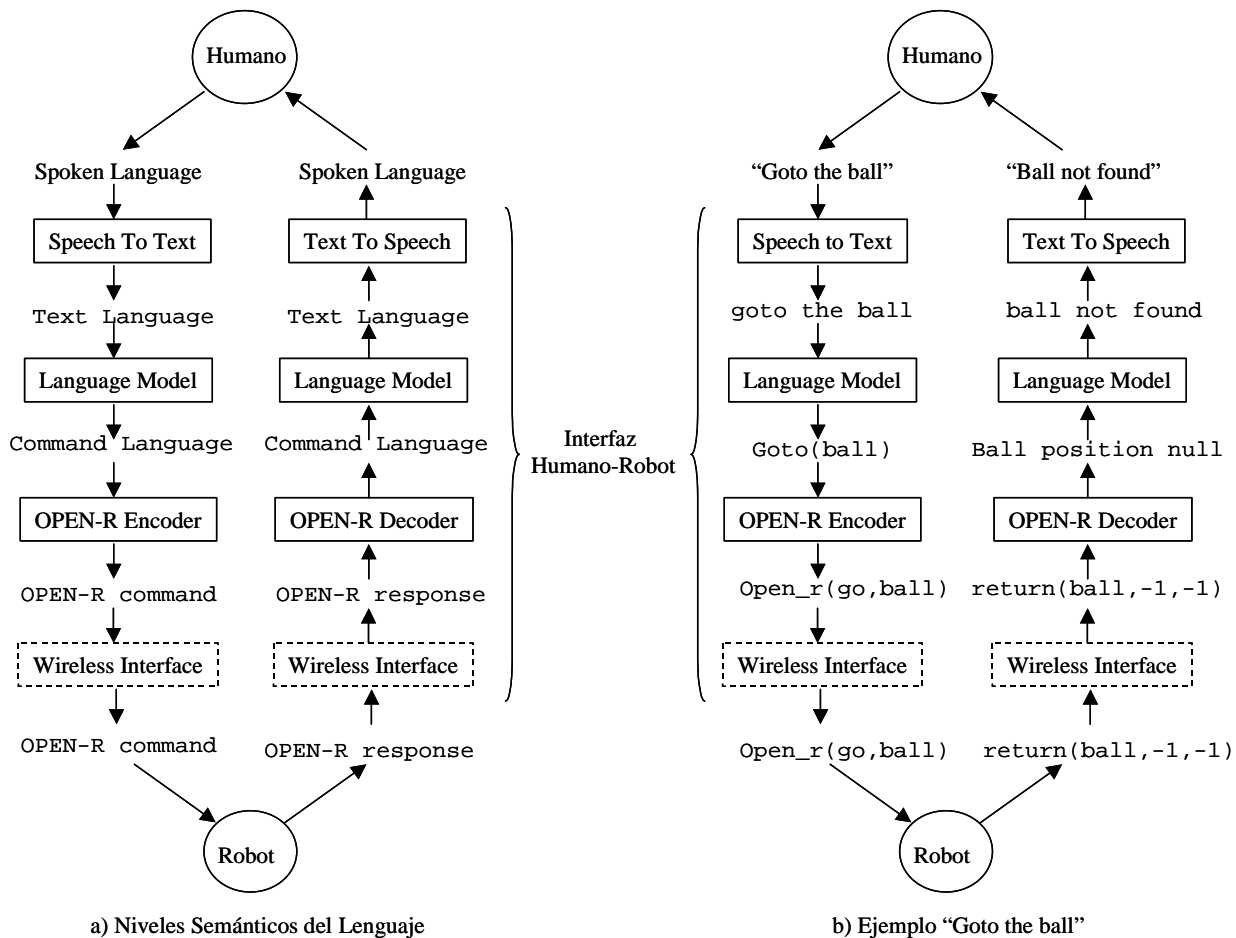
Basado en la configuración de la arquitectura anterior, y el modelo de diálogo de la Figura 5, definiremos nuevas interfaces entre el entrenador y la Interacción Humano-Robot (IHR), y lo más importante, las interfaces técnicas entre el IHR y el Aibo(s). Los elementos de la comunicación inalámbrica en el modelo de interfaz RAD ejecutarán dos funciones de administración de interfaz: Convertirán comandos de voz (definidos en 5.3.1) a comandos de Aibo en formato OPEN-R, y a su vez convertirán respuestas de Aibo en OPEN-R a oraciones comprensibles por un humano que luego serán escuchadas por un sintetizador.

##### 5.3.4.1 Comprensión y Niveles Semánticos del Lenguaje

El procesamiento del lenguaje natural en la interacción humano-máquina ha tenido sus mayores éxitos cuando el dominio de la interacción es restringido. Por ejemplo, en la planeación de reservación de viajes, un alcance limitado y bien definido de las posibles interacciones entre el humano y el agente permite el desarrollo de interfaces humano-máquina robustas para este dominio [20][21]. Explotaremos este tipo de enfoque en nuestro proyecto. El dominio de las posibles interacciones entre el humano y el robot será limitada y bien definida dentro del juego de fútbol, y por lo tanto nos permitirá evitar algunos de los problemas tradicionales asociados con la comprensión de lenguajes de alcance más general.

El sistema se basará en un procesamiento de información de múltiples niveles semánticos, como podemos apreciar en la Figura 7. Por ejemplo, si el entrenador envía el comando "Goto the ball", este comando será inicialmente procesado por el sistema *Speech to Text* para generar texto. A continuación, esto será entonces procesado por el modelo de lenguaje que lo reconocerá como una construcción propia del lenguaje, con el comando GOTO y el argumento THE BALL. Siguiendo la jerarquía, el sistema generará una construcción de comando de OPEN-R que será transmitida vía una interfaz inalámbrica al robot, en este caso el AIBO (se utiliza comunicación inalámbrica de tipo WiFi). Dentro del AIBO, este comando será procesado por funciones de interfaz del robot (programadas por nosotros aunque ligadas al ambiente de OPEN-R) las cuales involucran llamadas de alto nivel dentro del robot hasta llegar a niveles progresivamente inferiores consistiendo de llamadas a funciones propias de OPEN-R que permiten acceder, si fuera necesario, a los sensores y/o actuadores motrices específicos para poder realizar la acción solicitada por el humano.

Tanto el idioma de las interfaces como de los diversos niveles del lenguaje se hará en inglés como podemos apreciar de los ejemplos de construcciones mostrados en la Figura 7.



**Figura 7.** El diagrama muestra los niveles semánticos del lenguaje como parte de la Interfaz Humano-Robot. Se muestra en (a) los niveles generales y en (b) un ejemplo particular para "Goto the ball".

### 5.3.5 Probar módulos de software con interfaces simuladas

Validar (a) la habilidad del Aibo para recibir apropiadamente comandos y respuestas simuladas, y (b) la habilidad de la interfaz Humano-Robot para enviar y recibir comandos y respuestas simuladas, respectivamente.

### 5.3.6 Duplicar la configuración de un Aibo en Lyon para pruebas iniciales

El grupo de Lyon adquirirá un robot Aibo con fondos del grupo ACI Neuroscience Integrative et Computationelle.

### 5.3.7 Integración y pruebas en el ITAM

En el Laboratorio de Robótica del ITAM se cuenta con cinco robots AIBO y se está en proceso de adquirir otros cinco nuevos. Se harán pruebas con los Aibos actuales y nuevos.

### 5.3.8 Evaluación RoboCup

Se evaluará el sistema resultante en eventos de RoboCup jugando contra otros equipos en juegos amistosos. En México existen varios equipos que participan en la liga AIBO al igual que en Europa, Estados Unidos y Japón.

## 5.4 Lista de resultados a obtener:

A pesar de que este trabajo se hará en base a una interacción cercana entre los dos grupos de investigación, el Dr. Peter Dominey será responsable de los aspectos de lenguaje natural mientras que el Dr. Alfredo Weitzenfeld será responsable del control del robot.

#### A. Lenguaje Natural (Peter Dominey)

1. Especificación y prototipo de las construcciones comunicativas.
2. Especificación y prototipo del modelo de diálogo y protocolo de interacción.
3. Especificación y prototipo de los acuerdos de interfaz.

#### B. Control del robot (Alfredo Weitzenfeld)

1. Implantación construcciones comunicativas, modelo de diálogo y protocolo de interacción en software y hardware.
2. Pruebas de módulo de software con interfaces simuladas y AIBOs en tiempo real.
3. Evaluación en RoboCup.

### 5.5 Calendario del proyecto:

1. Reunión inicial en México – Descripción del Escenario: Mes 1; Duración 1 Semana; Objetivo: Trabajar en los ejemplos de simulación de las cuatro categorías de interacción, identificando las interfaces mayores, y validando la arquitectura nivel 1. Los entregables serán el Documento de Descripción del Escenario que incluye descripciones detalladas en UML (Lenguaje de Modelado Unificado) [22] de las interacciones de ambos lados Humano-Robot, indicando todas las interfaces, llamadas a nivel de sistema, etc.

2. Preparación del Documento de Descripción Funcional Versión 1: Meses 2-6; Duración 4 Meses; Objetivo: Generar una descripción funcional de la interfaz humano-maquina, con una explicación detallada de la estructura de los comandos verbales y el comportamiento correspondiente del robot. (5.4 A.1, A.2)

3. Desarrollo de Sistema Versión 1: Meses 3–6; Duración 3 Meses; Objetivo: Desarrollo de una versión preliminar del sistema que demuestre comunicación confiable entre humano y robot (comando usuario humano – respuesta sistema) para un subconjunto limitado del repertorio de interacción. (5.4 B.1)

4. Integración y Pruebas Versión 1: Meses 7-8; Duración 2 Meses; Objetivo: Integración y pruebas de la versión 1. (5.4 B.2)

5. Preparación del Documento de Descripción del Sistema e Interfaces Versión 2: Meses 7-10; Duración 4 Meses; Objetivo: Finalizar la arquitectura e interfaces incluyendo el mapeo de los diferentes tipos de interacción, y comandos y protocolo de control correspondientes predicado-argumento del robot. (5.4 A.3)

6. Desarrollo del Sistema Versión 2: Meses 9 – 12; Duración 3 Meses; Objetivo: Desarrollo de la versión final del sistema para los comandos y protocolos de control completos. (5.4 B.1)

7. Evaluación del Lenguaje Versión 2: Meses 11-12, Duración 2 Meses; Objetivo: Integración y pruebas de la versión final (5.4 B.2)

8. Integración y Pruebas Versión 2: Meses 13-14, Duración 2 Meses; Objetivo: Integración y pruebas de la versión final (5.4 B.2)

9. Pruebas RoboCup: Meses 15 – 18; Duración 4 Meses; Objetivo: Uso intenso del sistema. (5.4 B.3)

10. Valorización y Publicación: Meses 19-24; Esto incluirá la publicación de un breve documento de video científico que pueda ser usado por las agencias de financiamiento Mexicana y Francesa como demostración de tal proyecto de colaboración. (5.4 C)

## 5.6 Antecedentes de la cooperación:

Alfredo Weitzenfeld es el diseñador principal del Lenguaje de Simulación Neuronal NSL, un sistema para el modelado neuronal [11]. El trabajo en NSL se inició a principio de los 1990s en la Universidad del Sur de California (USC), donde un número extenso de modelos biológicos neuronales fueron desarrollados. Peter Dominey, en su tesis doctoral en USC desarrolló un modelo oculomotor de los monos utilizando el sistema NSL. Su trabajo se incluye en uno de los capítulos del libro de NSL [12]. Un proyecto recientemente financiado por parte del CONACYT en México (C03-42440) titulado “Robots Autónomos Neuroetológicos Visualmente Guiados: Un Enfoque de Middleware Adaptivo para Sistemas Móviles Embebidos Distribuidos” incluye una extensión al trabajo original de sistemas oculomotor de Peter Dominey [13].

## 5.7 Publicaciones:

- Dominey PF, Ramus F (2000) Neural network processing of natural language: I. Sensitivity to serial, temporal and abstract structure of language in the infant. *Lang. and Cognitive Processes*, 15(1) 87-127.
- Dominey, P.F. (2003a) Learning Grammatical Constructions in a Miniature Language from Narrated Video Events, Proceedings of the 25<sup>th</sup> Annual Meeting of the Cognitive Science Society, Boston.
- Dominey, P.F. (2003b) Learning Grammatical Constructions from Narrated Video Events for Human-Robot Interaction, Proc. IEEE Conf. On Humanoid Robotics, Karlsruhe.
- Dominey PF (2000) Conceptual Grounding in Simulation Studies of Language Acquisition, *Evolution of Communication*, 4(1), 57-85.
- Dominey PF, Hoen M, Lelekov T, Blanc JM (2003) Neurological basis of language in sequential cognition: Evidence from simulation, aphasia and ERP studies, (in press) *Brain and Language*.
- Weitzenfeld, A., Arbib, M.A., Alexander, A., 2002, The Neural Simulation Language: A System for Brain Modeling, MIT Press.
- Weitzenfeld, A., Gutierrez-Nolasco, S., Venkatasubramanian, N., 2003, Controlling Mobile Robots with Distributed Neuro-Biological Systems, Proc. AINS 2003, 2<sup>nd</sup> Annual Symposium on Autonomous Intelligent Networks and Systems, June 30 – July 3, Menlo Park, California.
- Weitzenfeld A., 2003, Embedded Mobile Systems: From Brain Theory To Neural-based Robots, Proc MED '03, 11<sup>th</sup> Mediterranean Conference on Control and Automation, June 17-20, Rhodes, Greece.
- Weitzenfeld A., Gutierrez-Nolasco S., and Venkatasubramanian N., 2003, MIRO: An Embedded Distributed Architecture for Biologically inspired Mobile Robots, Proc ICAR-03, 11<sup>th</sup> International Conference on Advanced Robotics, June 30 – July 3, Coimbra, Portugal.
- Arkin, R.C., Ali, K., Weitzenfeld, A., and Cervantes-Perez, F., 2000, Behavioral Models of the Praying Mantis as a Basis for Robotic Behavior, *Journal of Robotics and Autonomous Systems*, 32 (1) pp. 39-60, Elsevier.

## Significado del Trabajo

Esta colaboración se basa en áreas de investigación complementarias, en el ITAM en sistemas de robots colaborativos y en ISC en interfaces humano-robot. Esta colaboración es muy beneficiosa en los siguientes aspectos:

1. **Educación.** Muchos estudiantes graduados de instituciones Mexicanas incluyendo el ITAM han estado continuando sus estudios para grados avanzados en Estados Unidos y Europa. Actualmente, el ITAM ofrece programas de maestría de doble grado con École Nationale Supérieure des Télécommunications de Bretagne (ENSTB) e Institut National des Telecommunications (INT) ambos en Francia. Esperamos en un futuro tener una programa conjunto con el ISC donde estudiantes graduados del ITAM puedan continuar a grados más avanzados en áreas relacionadas a la robótica.
2. **Ciencia.** La robótica colaborativa y la interfaz humano-máquina son dos áreas importantes de desarrollo científico. Nuestra colaboración actual extenderá el trabajo científico al producir avances (y publicaciones) en estas dos áreas.
3. **RoboCup.** RoboCup es un área de desarrollo importante donde equipos de diferentes países se juntan en múltiples competencias contando con un evento mundial mayor una vez al año. En el ITAM comenzamos a competir hace dos años participando en dos abiertos americanos en dos categorías: robots pequeños (small-size) y AIBO. En la liga de robots pequeños obtuvimos el 3er lugar en el 2003 y el 2do en el 2004. En la liga AIBO comenzamos a participar recién este año. Este año se organizó en México la 3era competencia Latinoamericana de robótica donde se obtuvo el 1er lugar en small-size mientras que en AIBOs se jugó únicamente partidos de exhibición. Para el 2005 esperamos contar con un abierto Mexicano de RobCup y competir en el torneo mundial de RoboCup en ambas categorías. Estas participaciones han llamado la atención a muchos estudiantes en todo México para hacer proyectos y tesis en el Laboratorio de Robótica del ITAM. Los periódicos y la televisión en México han seguido con gran interés estas participaciones.
4. **Tecnología.** La interfaz humano-máquina y su uso en el control y monitoreo de múltiples robots colaboradores tiene significado más allá del fútbol. Una de las mayores aplicaciones de la tecnología de RoboCup es búsqueda y rescate en desastres de gran escala. RoboCup inició un proyecto RoboCupRescue [9] para específicamente promover la investigación en temas con significado social relacionados a la robótica de búsqueda y rescate urbano (USAR) [8,14]. Esta tecnología comienza a tener un impacto importante en países donde los terremotos son comunes y en respuesta a otros desastres relacionados como el caso de las torres gemelas de Nueva York [10]. Esperamos que el trabajo de esta propuesta tenga cada vez mayor importancia en la sociedad, en apoyo a aplicaciones donde se arriesga la vida al igual que otras.

## Bibliografía

- [1] AIBO official web site. Url: [www.aibo.com](http://www.aibo.com).
- [2] Open R SDK official web site: <http://openr.aibo.com/openr/eng/index.php4>.
- [3] RoboCup official web site: <http://www.robocup.org>.
- [4] Chen, et al. "A Description of the rUNSWift 2003 Legged Robot Soccer Team". Australia, 2003. Url: <http://www.cse.unsw.edu.au/~robocup/teamDesc2003.pdf>.
- [5] Chen, et al. "Rise of the AIBOs III –AIBO Revolutions". Australia, 2003. Url: <http://www.cse.unsw.edu.au/~robocup/report2003.pdf>.
- [6] Mitsunaga, Toichi, Izumi and Asada. "Baby Tigers 2003: Osaka Legged Robot Team" Japan 2003. Url: <http://www.er.ams.eng.osaka-u.ac.jp/robocup/BabyTigers/BabyTigers-TechReport-2003.pdf>.
- [7] Veloso, Winner, Lenser, Bruce, and Balch, "Vision-servoed localization and behavior-based planning for an autonomous quadruped legged robot", In Proceedings of AIPS-2000, Breckenridge, April 2000.
- [8] Rescue Robotics, IEEE Robotics & Automation Magazine, 9 (3), September 2002.
- [9] RoboCupRescue, Urban Search and Rescue Robot Competitions, 2004 (<http://www.isd.mel.nist.gov/projects/USAR/competitions.htm>).
- [10] Murphy, R., Rescue Robotics for Homeland Security, Special Issue on Emerging Technologies for Homeland Security, Communications of the ACM, 47 (3) 66 – 68, March 2004.
- [11] Weitzenfeld, A., Arbib, M.A., Alexander, A., 2002, The Neural Simulation Language: A System for Brain Modeling, MIT Press.
- [12] Dominey, P, Arbib, M.A., and Alexander, A., 2002, The Modular Design of the Oculomotor System in Monkeys, in The Neural Simulation Language: A System for Brain Modeling, Eds. Weitzenfeld, A., Arbib, M.A., Alexander, A., MIT Press.
- [13] Dominey P.F., and Arbib M.A., A Cortico-Subcortical Model for Generation of Spatially Accurate Sequential Saccades, Cerebral Cortex, 2:153-175, 1992.
- [14] Orfinger, B., Robot Responders at WTC Site Fit Into Tight Spaces, Disaster Relief, Oct 2001, (<http://www.disasterrelief.org/Disasters/011015robots/>).
- [15] Siskind JM (2001) Grounding the lexical semantics of verbs in visual perception using force dynamics and event logic. *Journal of AI Research* (15) 31-90.
- [16] Steels, L. and Baillie, JC. (2002). Shared Grounding of Event Descriptions by Autonomous Robots. *Robotics and Autonomous Systems*, 43(2-3):163--173.
- [17] CMU Communicator: <http://www.speech.cs.cmu.edu/Communicator/>.
- [18] CSLU Toolkit <http://www.cslu.ogi.edu/toolkit/index.html>.
- [19] Dominey PF, Boucher J-D, Inui T (2004) Building an Adaptive Spoken Language Interface for Perceptually Grounded Human-Robot Interaction, IEEE Conf. On Humanoid Robotics 2004, Los Angeles.
- [20] Allen, J., Byron, D., Dzikovska, M., Ferguson, G., Galescu, L., and Stent, A., (2001) "Towards Conversational Human-Computer Interaction," *AI Magazine*.
- [21] Ferguson G., and Allen, J.F., (1998) TRIPS: An Integrated Intelligent Problem-Solving Assistant, Proceedings of the Fifteenth National Conference on AI (AAAI-98), Madison, WI, 26-30 July.
- [22] Weitzenfeld, A., (2004) Ingeniería de Software Orientada a Objetos con UML, Java e Internet, Thomson Learning, México.