
DESARROLLOS RECIENTES EN ROBOTICA EDUCACIONAL (*)

(*) Ponencia presentada al Taller Andino de Sistemas Expertos y Robótica. Agosto 1989.

Traducción autorizada por el autor, por Juan Guillermo Arango Vásquez, Jefe Dirección Servicios Universitarios EAFIT.

MAURICE BELANGER, CIRADE

- B.Sc. Biología y Química de la Universidad de New Hampshire, USA.
- Ed.D. Ciencias y Educación Matemáticas de la Universidad de Harvard.
- Estudios de Postdoctorado en Epistemología Genética de la Universidad de Ginebra.
- Jean-Baptiste LaPalme, Département de Mathématiques et Informatique, Université du Québec a Montreal.

INTRODUCCION

Durante la última década ha tenido lugar en los círculos públicos una permanente discusión sobre las "nuevas tecnologías". Mucho énfasis se ha dado al papel que las nuevas tecnologías están desempeñando en la transformación del ámbito y los sitios de trabajo, al impacto sobre la economía mundial y a la influencia, frecuentemente de una manera sutil, en la vida diaria de millones de individuos. En una forma muy general, existe una situación similar a la ocurrida en muchas naciones durante el siglo diecinueve, cuando los desarrollos en ciencia y tecnología tuvieron una influencia dramática sobre las transformaciones socioculturales.

Un área en donde los nuevos desarrollos tuvieron un profundo impacto en el siglo diecinueve fue la Educación. El tradicional "currículo de artes liberales" fue gradualmente modificado para incluir más ciencia y matemáticas, y la educación vocacional y profesional llegó a ser parte del programa de la escuela secundaria. Pasaron muchas décadas antes de que el currículo, los textos y otros materiales pedagógicos comenzaran a reflejar énfasis sobre la preparación del joven para la vida en una sociedad de rápida industrialización, donde la ciencia y la tecnología estaban convirtiéndose en fuerzas conductoras. Durante los últimos cuarenta años, los computadores y los conceptos de las ciencias de la información han llegado a ser nuevas fuerzas del cambio sociocultural y las escuelas a nivel preuniversitario están comenzando muy lentamente a reflejar estas innovaciones. Sin embargo, una gran cantidad de trabajo está por hacerse en la elaboración de material pedagógico y en la enseñanza y el aprendizaje de estrategias apropiadas para los estudiantes no especialistas, quienes de todas formas necesitarán tener un nivel mínimo de conocimiento de los nuevos conceptos para ser capaces de operar y trabajar en el siglo veintiuno.

Es en este amplio contexto que un número de individuos y grupos en Québec han estado trabajando durante los últimos años, para desarrollar una serie de materiales pedagógicos y estrategias de aprendizaje incluyendo aplicaciones pedagógicas de las ciencias de la computación. Muchos de estos esfuerzos han sido desarrollados con la colaboración de APO-Québec (Aplicaciones Pedagógicas del Ordenador), una organización creada por el Ministerio de Educación de Québec. En este breve artículo se describen algunos aspectos seleccionados de la

investigación y el desarrollo emprendido para introducir estudiantes a las ideas básicas de la robótica. En particular, se describen algunos trabajos desarrollados durante los últimos cinco años con la participación de Stéphane Bélanger.

OBJETIVOS

Estudiantes, probablemente bajo la influencia de la televisión, el cine y los fabricantes de juguetes, han desarrollado algunas ideas fantásticas relacionadas con los robots, similares a las que tiene gran parte del público en general. Ellos frecuentemente atribuyen a los robots capacidades que están más allá de los más recientes avances en la inteligencia artificial; todavía estos estudiantes tienen poco o ningún conocimiento de muchas nociones elementales que son los conceptos básicos que soportan los dispositivos existentes. Los estudiantes se dan cuenta que de alguna manera un computador está involucrado, pero aun así atribuyen mucha característica misteriosa y mágica a los computadores hasta que desarrollen por sí mismos una considerable cantidad de experiencias con ellos.

Hace cerca de cinco años que se inició el desarrollo de material pedagógico a costos razonables para que pudiera ser usado en escuelas primarias y secundarias en la iniciación de los estudiantes a las ideas elementales de la robótica. En ese tiempo estaban disponibles una serie de elementos de construcción, como: motores, luces, interruptores, etc. que podían ser utilizados para elaborar varios dispositivos. Algunas compañías que producen elementos de construcción, tales como Fischer-Technik en Alemania, tenían interfases disponibles que podían ser conectadas a un microcomputador. Escribiendo un programa en BASIC varios de estos dispositivos contruídos con estos materiales podían ser controlados. Se probaron varias interfases comerciales de diferentes compañías y se encontró que mientras éstas trabajaron muy bien para aplicaciones sencillas ninguna de ellas permitía trabajar en el nivel de complejidad deseado. En este momento se presentaron una serie de especificaciones a Caméléon Inc. para desarrollar una interfase que satisficiera los requerimientos. Todos los trabajos desde ese entonces han sido desarrollados con las interfases de Caméléon usando la versión para el computador Apple II o la del computador IBM compatible.

Una dificultad que se encontró experimentando con BASIC es que este lenguaje es demasiado lento e

incómodo para ser utilizado como un lenguaje de control. Un lenguaje de computador que trabaja muy bien para dispositivos simples robotizados es LOGO, el cual muchos niños de Québec aprenden en la escuela. Sin embargo, trabajando con estudiantes de escuela secundaria, quienes elaboran construcciones más complejas, LOGO posee algunas características que lo hacen poco confiable en situaciones de control donde el tiempo es un factor crítico. (Esta dificultad se deriva de la manera como LOGO manipula la recursividad). Para superar estas dificultades uno de nosotros, J. B. LaPalme, desarrolló durante dos años un nuevo lenguaje (denominado Android en este artículo) el cual, recientemente está disponible en francés.

Después de este período inicial de desarrollo de software y de hardware se obtuvieron las herramientas necesarias para abordar el desarrollo y la investigación en robótica educacional con los estudiantes, con los siguientes objetivos básicos:

1. Dar acceso a algunos de los conceptos elementales de control a estudiantes entre once y diecisiete años y a adultos no especializados en este campo.
2. Desarrollar material pedagógico en robótica dentro de un marco de solución de problemas.
3. Explorar el aprendizaje y utilizar el lenguaje Android, el cual es modelado sobre lenguaje natural, para la programación del computador en los dispositivos construídos por los estudiantes.
4. Mantener el costo de los equipos y los materiales dentro de las limitaciones presupuestales de las escuelas.

PROGRAMACION E INTERFAZ CON EL COMPUTADOR

En el trabajo con los estudiantes de secundaria durante los últimos dos años se utilizaron una diversidad de materiales disponibles en el mercado. En particular se utilizaron materiales de Fischer-Technik de Alemania, Lego de Dinamarca y Robotrix de los Estados Unidos. Adicionalmente se trató de utilizar materiales fácilmente adquiribles a bajo costo en las tiendas de computadores y en las casas proveedoras de elementos electrónicos.

Después de que un estudiante o grupo de estudiantes han construído un dispositivo que puede estar

compuesto de motores, luces, interruptores, potenciómetros, etc., los componentes pueden ser conectados a la interfaz de Caméléon por medio de una caja conectora elaborada por la misma compañía. En el trabajo con los estudiantes se planeó poder trabajar desde un nivel muy simple de solución de problemas involucrando solamente unos pocos componentes, hasta un nivel de complejidad considerable. Se podía haber trabajado con los estudiantes con lenguajes tales como Basic, Pascal, C o Logo pero la experiencia en la enseñanza de lenguajes de computación a nivel universitario nos llevó a detectar dificultades encontradas por los estudiantes cuando ellos están confrontados con la tarea de escribir un programa para resolver un problema específico. Cada lenguaje de computación tiene sus propias convenciones, reglas de sintaxis, etc. Cuando un estudiante escribe un programa para resolver un problema, el o ella debe transcribir (o traducir) el problema en términos de las reglas y las convenciones del lenguaje de computación que está siendo utilizado. En la experiencia docente se ha notado que a veces los estudiantes llegan a estar tan enredados en el trabajo con el problema y su solución en términos del código del lenguaje utilizado, que pierden todo el sentido del problema originalmente planteado.

Una de las ideas fundamentales de J. B. LaPalme, para el diseño de Android, fue escribir el lenguaje de tal forma que el estudiante pudiera usar las reglas y las convenciones de su lenguaje natural en el proceso de pensar y escribir un programa de computador para resolver un problema específico. Android ha sido diseñado de tal forma, que el usuario enfrentado con la solución de un problema utilizando este lenguaje necesita pensar en términos de las palabras y las frases del lenguaje natural que el tendrá que crear para resolverlo. A pesar de que es imposible en este artículo entrar en detalles de la programación con Android para resolver problemas, una serie de ejemplos pueden dar una idea de el enfoque general del lenguaje.

EJEMPLO 1: UNA PALABRA EQUIVALENTE A UN PROGRAMA

Un ejercicio favorito de los estudiantes es el de construir un carro que pueda moverse hacia adelante (FORWARD) y hacia atrás (BACKWARD) y cuyas ruedas delanteras puedan girar a la izquierda (LEFT) o a la derecha (RIGHT). Este carro puede ser construído con elementos de Fischer-Technik o LEGO usando dos motores conectados a las salidas

(EXITS) sobre la caja de conectores, tal como se ilustra en la figura No. 1.

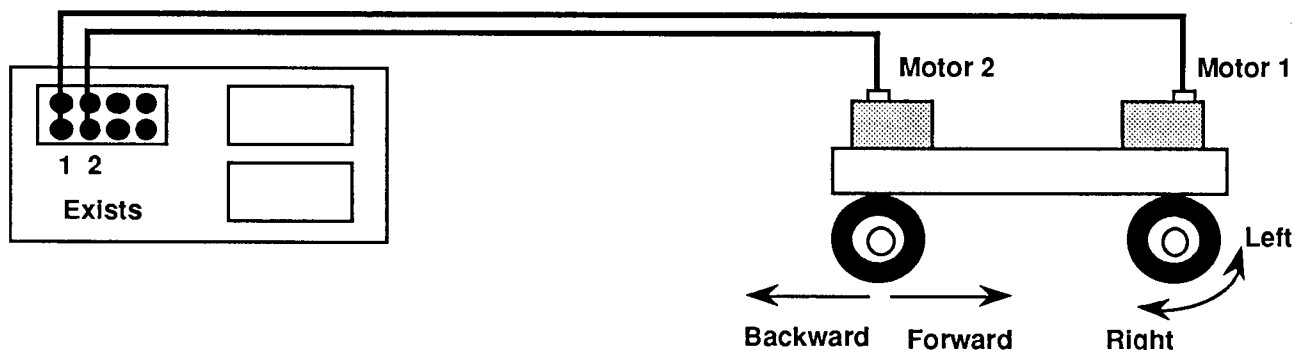


FIGURA No. 1

Lo que podría ser altamente deseable para programar este carro es la posibilidad de crear nuevas palabras tales como BACKWARD, FORWARD, RIGHT y LEFT. Una vez creadas, al teclear una de estas palabras en el computador se activaría el carro para ejecutar la acción correspondiente. Para ser capaz de hacer ésto se necesita un lenguaje de computador que sea extendible, esto es, un lenguaje que ofrezca la posibilidad de definir nuevas palabras. Para el caso del carro cada palabra representa una acción. LOGO, C y FORTH vienen inmediatamente a la mente como lenguajes de computador que ofrecen estas posibilidades. Con niños de más corta edad se ha estado experimentando con LOGO para que nuevas palabras (procedimientos) puedan ser fácilmente creadas. Sin embargo, debido a que también fue la intención trabajar con estudiantes de mayor edad sobre problemas relativamente complejos, el "ciclo de recolección de basuras" de LOGO y otras restricciones hicieron que este lenguaje no fuera apto para los propósitos. El lenguaje ideal para usar en procesos complejos de control habría sido FORTH, pero encontrar estudiantes jóvenes para programar en este lenguaje es imposible. Estas y otras consideraciones condujeron a percibir la necesidad de crear un nuevo lenguaje, que tuviera las características muy potentes de FORTH, pero al mismo tiempo fuera accesible a estudiantes principiantes mediante su semejanza con la forma del lenguaje natural.

A pesar de que el propósito no es el de conocer cómo programar en Android (Recientemente ha sido elaborado un manual de instrucción en francés para este propósito) se puede al menos ilustrar la definición de una nueva palabra con relación de nuevo al ejemplo del carro de la figura No. 1. Android

consiste de una serie de palabras primitivas. Algunas de estas primitivas están orientadas a la comunicación con la interfase Caméléon a través de la caja de conectores. Por ejemplo, las primitivas POS (numero), NEG (numero) y NUL (numero) colocarán respectivamente: un voltaje (POSitivo) en la salida (EXIT) correspondiente al número tecleado, un voltaje de polaridad opuesta (NEGativo) o anulará (NULO) el voltaje. En el caso del carro de la figura 1 se asume que el mecanismo de engranajes es tal que POS 1, movería el carro hacia adelante (FORWARD). Usando la primitiva DEF (DEFina) y END se podría crear una nueva palabra FORWARD, así:

```
DEF FORWARD
POS 1
END
```

La sintaxis para crear una nueva palabra comienza con la primitiva DEF, seguido por la nueva palabra, seguido por la declaración de la definición y finalmente la primitiva END indica el fin de la definición. Las otras tres palabras podrían ser definidas como sigue:

```
DEF BACKWARD NEG 1 END
DEF RIGHT POS 2 END
DEF LEFT NEG 2 END
```

Este ejemplo sólo demuestra como una nueva palabra puede ser adicionada al vocabulario de Android, pues la situación construida en la figura No. 1 es demasiado simple para ser de interés. Una variación mas interesante sería la de colocar un interruptor (SWITCH) en la parte frontal del carro y un segundo interruptor en la parte posterior del mismo y conectar estos interruptores a las entradas

identificadas como 1 y 2 respectivamente, tal como está ilustrado en la figura No. 2.

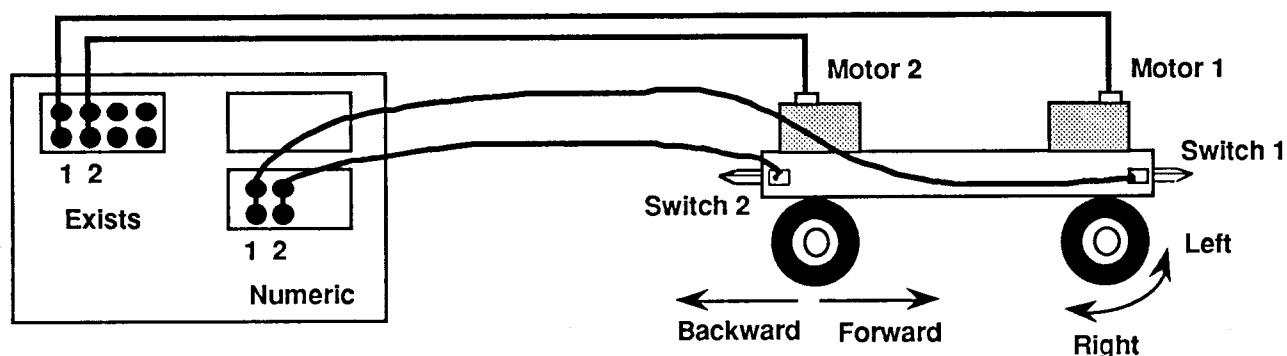


FIGURA No. 2

Este dispositivo modificado (el carro con interruptores) introduce a los estudiantes al mundo de los sensores y a entender como el estado de un sensor puede ser usado para controlar la dirección de rotación de un motor. Android incluye un conjunto completo de instrucciones primitivas para leer el estado de entradas numéricas y análogas que entran a la caja de conectores Camélón, un completo conjunto de primitivas condicionales (tales como IF, WHILE, UNTIL, etc) y las funciones "Booleanas" usuales (AND, OR, NOT) todo esto con el fin de dar a los estudiantes la capacidad de construir y programar dispositivos para resolver problemas elementales como los siguientes:

PROBLEMA: Construir un carro que se mueva hacia adelante y hacia atrás, y que las ruedas delanteras giren a la derecha y a la izquierda. Colocar un interruptor en la parte delantera del carro. Escribir un programa en Android que pare el carro cuando choque con un obstáculo, retroceda una distancia de un pie, las ruedas cambien de dirección, el carro se mueva hacia adelante un cuarto de círculo, entonces las ruedas giren de tal forma que el carro se mueva en una línea recta y se mantenga rodando hasta que choque con otro obstáculo, etc.

PROBLEMA: Construir un semáforo usando luces roja, amarilla y verde conectadas a la caja de conectores. Escriba un programa para controlar el tráfico en una sola vía.

Variación: Adicione un interruptor tal que el peatón pueda parar el tráfico para atravesar la calle.

Variación: Construya un segundo semáforo y escriba un programa que controle el tráfico en un cruce de dos vías.

La solución de estos problemas implica analizar la máquina misma para ver en que componentes puede ser dividida. Adicionalmente se debe analizar que nuevas palabras (comandos) sería útil crear. Como en el lenguaje natural, las palabras pueden estar encadenadas en frases, y se necesita pensar por adelantado que frases serían necesarias o útiles en la solución del problema. Este aspecto puede ser ilustrado con el siguiente ejemplo.

EJEMPLO 2: FRASES COMO PROGRAMAS

El primer ejemplo ha ilustrado como nuevas palabras pueden ser creadas usando primitivas de Android. Una vez definidas, estas palabras llegan a ser parte del vocabulario para controlar un dispositivo en particular. Este es el primer paso. Una vez que un conjunto de palabras han sido creadas, estas palabras pueden ser utilizadas en la escritura de frases similares al lenguaje natural usando otras primitivas de Android. El siguiente ejemplo ilustra cómo programas en Android pueden tomar la forma de frases del lenguaje natural. Considere el siguiente problema que nosotros hemos planteado a los estudiantes de secundaria quienes están en el proceso de aprender Android y quienes han tenido alguna experiencia con motores, luces, interruptores y celdas fotoeléctricas en lecciones previas.

PROBLEMA: Usando bloques de LEGO construya una puerta que abra y cierre por medio de un motor (Ver figura No. 3). Se necesita construir

el mecanismo que haga ésto. Cuando la puerta está cerrada el interruptor 2 está presionado. Cuando la puerta está abierta el interruptor 1 está presionado. Coloque celdas fotoeléctricas y luces sobre cualquier lado de la puerta. Escriba una palabra en Android, que controle la apertura y el cierre de la puerta cuando la gente camine al frente de la celda fotoeléctrica, como en un supermercado y un aeropuerto. Habrá que descifrar cómo conectar el motor, las luces, los interruptores y las celdas fotoeléctricas a la caja de conectores Caméléon.

Después de construir un mecanismo que trabaje bien, el primer paso para escribir un lenguaje de aplicación en Android es crear una serie de palabras que sean apropiadas para el dispositivo específico. Esa serie de palabras podría ser la siguiente:

DOOR	CLOSED	CLOSES	OPEN
OPENS	EYE1	EYE2	EYE
EITHER	THE	IS	

Si estas palabras son consideradas con las siguientes primitivas de Android: IF, AND, OR, entonces las siguientes frases pueden ser escritas:

IF THE DOOR IS CLOSED AND EITHER EYE IS CUT,
THE DOOR OPENS.
IF THE DOOR IS OPEN AND EITHER EYE IS CUT,
THE DOOR CLOSES.

A pesar de que estas dos frases parecen ser sólo una descripción de la situación y su lógica, las dos frases son de hecho el programa. Todo lo que se necesita para operar este modelo de una puerta automática es colocar estas dos frases en la definición de una palabra y la ejecución de esa palabra hará que la puerta funcione. (Podría también ser necesario descifrar la manera de prender las dos luces).

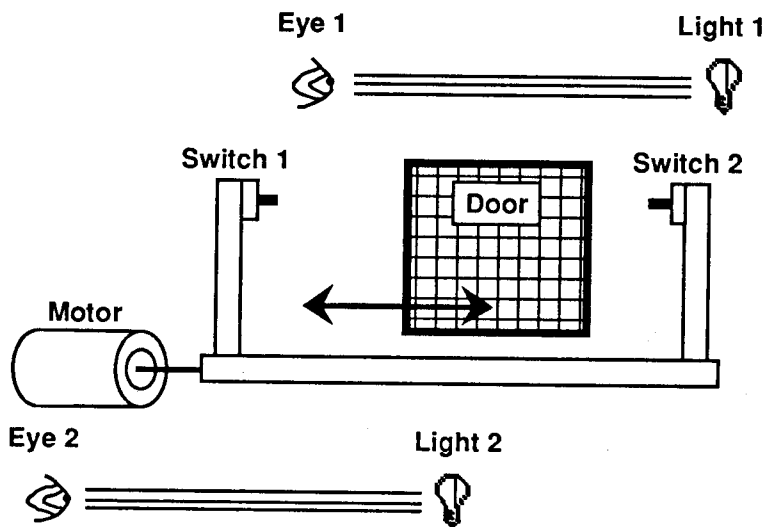


FIGURA No. 3

Esta característica de Android de permitir la escritura de un programa o partes de un programa en la forma de frases del lenguaje natural es motivada por la observación previamente discutida relacionada con las dificultades del estudiante al perder el sentido del problema si él está condenado a traducir la solución del mismo al código específico del lenguaje de programación. Esta es una hipótesis que requiere mayor confirmación. Sin embargo, el trabajo con los estudiantes durante los dos últimos años ha reafirmado la creencia de que si el problema de programación está cercano al propio pensamiento del estudiante en lenguaje natural (francés, inglés,

español, etc.) él es más capaz de pensar en el problema y su solución. A pesar de que se intenta canalizar más investigación sobre este punto durante los dos o tres próximos años, los resultados obtenidos hasta ahora con estudiantes principiantes son alentadores. Un tercer ejemplo ilustra otra característica de Android.

EJEMPLO 3: ANDROID COMO UN LENGUAJE ORIENTADO A OBJETOS

Una tercera característica de Android que es importante resaltar, es que Jean-Baptiste LaPalme

también diseñó el lenguaje en la forma de un lenguaje orientado a objetos. Esto significa que Android tiene una estructura jerárquica, posee herencia, puede pasar mensajes y posteriormente tendrá la habilidad de definir clases. Aunque, que es y que no es un lenguaje objeto es todavía materia de considerable discusión y controversia, parece existir algún consenso sobre las características mencionadas. Aquí sería importante simplemente demostrar por medio de otro ejemplo el uso de Android como un lenguaje orientado a objetos y como la solución a los problemas puede ser significativamente simplificada. El ejemplo involucra la programación de un brazo de entrenamiento robotizado fabricado por Fischer-Technik disponible en el mercado. Se ha modificado el alambrado de este robot para que los motores, contadores e interruptores puedan ser conectados a la caja de conectores de Camélion. El siguiente diagrama simplificado ilustra los componentes básicos del robot de entrenamiento.

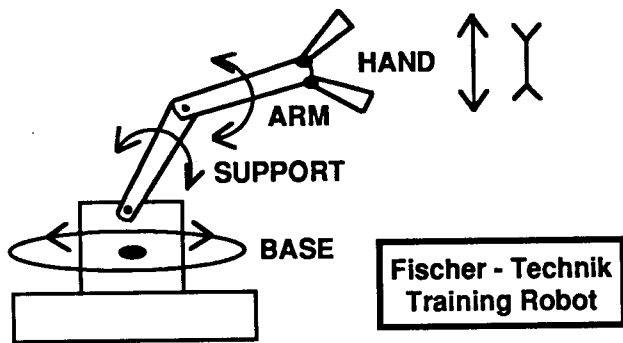


FIGURA No. 4

Se puede considerar que el robot de entrenamiento está compuesto de cuatro partes: una base, un soporte, un brazo y una mano. Cada uno de estos componentes es movido por un motor bidireccional. Hay tres contadores asociados respectivamente con la base, el soporte y el brazo. Finalmente hay cuatro interruptores, uno por cada componente básico que puede ser utilizado para iniciar la posición de las secciones principales. Cada componente puede ser visto como un objeto, donde cada objeto posee su propio conjunto de palabras. Se pueden visualizar los objetos y su respectivas acciones posibles (ver Figura No. 5).

En Android cada componente puede ser definido como un objeto y a cada objeto se le pueden definir palabras. El objeto de más alto nivel es el robot y si una palabra está asociada a él, todos los cuatro objetos del siguiente nivel en la jerarquía heredarán la habilidad para ejecutar esa palabra o comando. Con el fin de ser capaces de escribir frases para dirigir el robot otras palabras pueden ser definidas, tales como: MOVES, MOVE TO, MOVE BY, THE, etc. Una vez que los objetos y sus respectivas palabras han sido definidas, al igual que otras palabras necesarias para escribir frases, entonces un lenguaje de aplicación ha sido creado y puede ser utilizado para controlar el robot de entrenamiento de Fischer-Technik a través de la escritura de frases bajo la formas del lenguaje natural. Debido a que los contadores pueden ser leídos para dar un valor de la posición de la base, el soporte y el brazo, entonces cada punto en el ámbito del robot puede

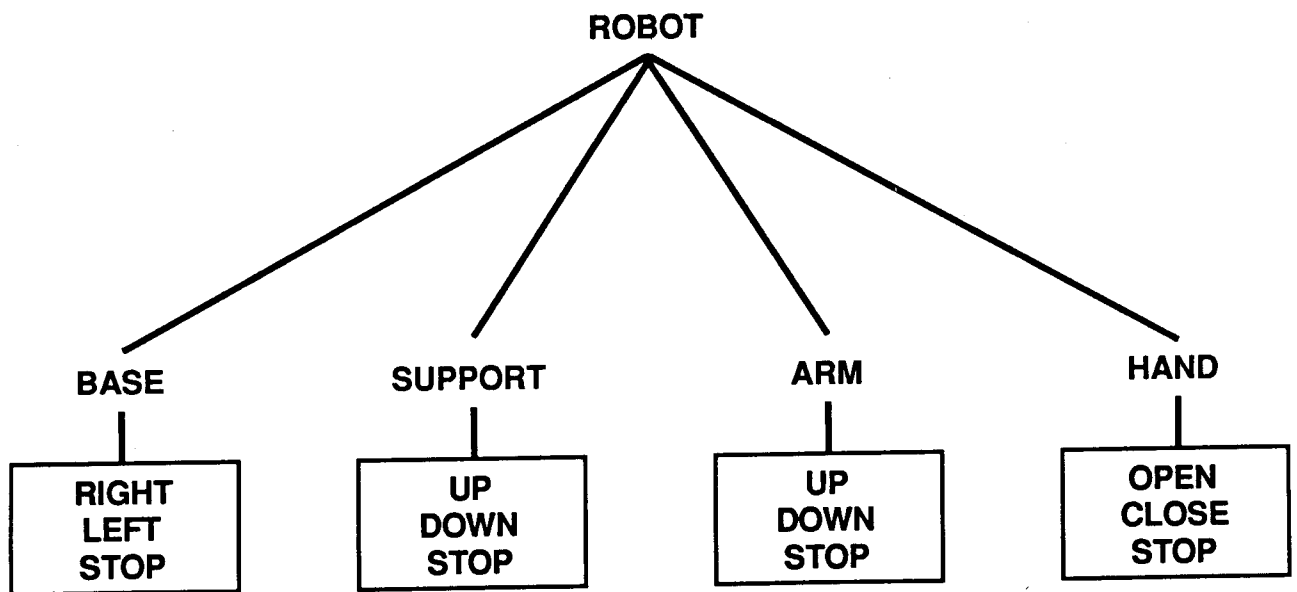


FIGURA No. 5

ser referenciado. En 1988 se experimentó con niños de once y doce años presentándoles los cuatro objetos básicos del robot y las palabras que cada objeto ejecutaría o entendería. En un tiempo muy corto (aproximadamente una hora) los niños llegaron a ser bastante expertos en la programación del robot para mover objetos en la solución de problemas espaciales sencillos y en la invención de sus propios problemas y soluciones.

En la medida en que los problemas planteados a los estudiantes llegan a ser más complejos y requieren dispositivos compuestos de un mayor número de componentes físicos (motores, luces, sensores, etc.), se ve claramente que la habilidad para usar las características de un lenguaje orientado a objetos reduce sustancialmente la complejidad de crear un lenguaje de aplicación para un dispositivo específico. El último ejemplo dará una ilustración de este punto.

EJEMPLO 4: UN PROYECTO COMPLEJO DE LOS ESTUDIANTES-UN TEATRO ROBOTIZADO

En el otoño de 1988 se inició el proyecto más complejo abordado hasta la fecha con los estudiantes de secundaria con edades entre los trece y diecisiete años. El objetivo básico fue construir un teatro de títeres que permitiera la ejecución de una representación completamente controlada por computador mediante palabras y frases escritas en Android. Doce estudiantes estuvieron involucrados en el diseño y construcción del teatro robotizado durante un período de ocho meses. Los estudiantes decidieron que los componentes básicos del teatro serían:

1. Un telón que abre y cierra.
2. Cuatro páneces escenográficos que podrían ser subidos y bajados.
3. Una variedad de luces en diferentes colores para iluminar el escenario.
4. Tres robots (títeres) que ejecutarían una variedad de acciones.

Posteriormente se adicionaron tres requerimientos adicionales:

5. Los títeres serían capaces de hablar.
6. La música sería colocada en cualquier momento durante la ejecución de la representación.

7. Dos computadores Apple II serían utilizados y podrían comunicarse entre sí.

Durante los meses de octubre a diciembre de 1988 los doce estudiantes se dividieron en cuatro equipos: Tres estudiantes para telón y páneces, tres estudiantes para luces, cuatro estudiantes para títeres y dos estudiantes para comunicaciones. Los equipos experimentaron con una variedad de prototipos que ellos mismos construyeron y luego escribieron en Android programas para controlarlos. Desde febrero a abril de 1989 un prototipo de teatro completo fue construido y la primera demostración fue presentada en mayo. El teatro en su forma actual está compuesto de diecinueve motores, cuatro juegos de luces, quince potenciómetros, seis celdas foto-eléctricas y dos interruptores. Dos computadores Apple II son utilizados, cada uno con dos tarjetas de interfase con Caméléon. Esto es un proyecto de un nivel intermedio de complejidad en su construcción y programación para estudiantes principiantes de secundaria. Sin embargo, una vez que las palabras y las frases para operar el teatro han sido definidas (el lenguaje de aplicación para el teatro), la programación de una representación es cuestión de escribir en el computador palabras o frases para llevar a cabo las series de acciones requeridas por la trama o el manuscrito de la representación. Por ejemplo, las siguientes frases en Android podrían representar la escena de apertura de una representación:

OPEN DE CURTAIN.
TURN ON THE RED LIGHT.
ZOE GOES ON STAGE. (Zoe es el nombre de uno de los títeres)
ZOE FACES THE PUBLIC.
ZOE BENDS DOWN.
ZOE RAISES UP.
ZOE TURNS RIGHT.
ZOE GOES OFF THE STAGE.
TURN OFF THE RED LIGHT.
CLOSE THE CURTAIN.

Este podría ser el programa real que controla las acciones. Con la simple lectura de este programa uno podría fácilmente describir qué acciones tienen lugar: una cortina que se abre; una luz roja que se enciende; un actor entra al escenario; el actor se inclina; el actor se vuelve y camina fuera del escenario; la luz roja se apaga y la cortina se cierra. El punto esencial es que una vez el lenguaje de aplicación queda definido, la programación de una representación llega a ser una cuestión de escribir

frases con la formas del lenguaje natural y no es un problema de recordar una serie larga y sin mucho sentido de pseudocódigo en un lenguaje de programación convencional.

En el año escolar 1989-90 se intenta trabajar con niños de primaria para que ellos primero escriban una representación, diseñen los vestidos de los robots, diseñen y pinten los escenarios y finalmente programen en Android la representación que escribieron. Se espera que con poco entrenamiento los niños serán capaces de usar el lenguaje de aplicación para ejecutar representaciones usando el teatro robotizado.

Un aspecto de Android que todavía no ha sido mencionado, pero que es crítico para operar el teatro robotizado (y en general para la robótica), es la capacidad de varios componentes de un dispositivo o de varios dispositivos diferentes para llevar a cabo acciones simultáneas. La primitiva de Android SIMULTANEOUS permite que varias acciones que han sido programas independientemente sean ejecutadas simultáneamente, simulando un procesamiento en paralelo. En el caso del teatro robotizado, usando SIMULTANEOUS el computador es responsable de administrar las acciones simultáneas de todos los títeres. Para un mayor número de tareas, esta característica es esencial y su inclusión en Android permite al usuario simular muchas tareas del mundo real.

COMENTARIOS FINALES

Los ejemplos anteriores pueden parecer que están muy distantes de la concepción profesional de un robot industrial, pero de acuerdo con lo

mencionado en la introducción, el objetivo no es entrenar profesionales, sino permitir a los estudiantes a nivel de primaria y secundaria el aprendizaje de algunos conceptos e ideas fundamentales en el campo de la robótica. Así como se cree que los estudiantes deben tener algunos conocimientos elementales en biología para entenderse a sí mismos y entender su ambiente y muchos de los problemas ecológicos que son de alcance mundial, así mismo creemos que los estudiantes de hoy deben también poseer algunas ideas elementales de la robótica y el control para entender la tecnología presente y futura.

Aunque aún no se ha trabajado con educadores involucrados con el entrenamiento de profesionales o con trabajadores a quienes les gustaría tener una introducción elemental a las ideas de la robótica y el control, se considera que los materiales descritos aquí podrían fácilmente ser adaptados a la población adulta.

Hasta el presente las primitivas del lenguaje Android están escritas en francés, pero hacia finales de 1989 habrá una versión disponible en inglés. También está disponible un manual de instrucción en Android en francés orientado a estudiante de primaria y secundaria. Una traducción de este manual al inglés se espera también que esté disponible hacia finales de 1989.

Finalmente, se podría anotar que Android fue escrito en lenguaje FORTH. Android podría haber sido escrito para cualquier lenguaje natural que pudiera ser entrado a través del teclado de un computador. En el presente el desarrollador (J.B. LaPalme) está considerando seriamente la preparación de una versión futura de Android en español.