

# Un Marco Formal para la Programación Implícita de Robots

**Jorge Elías Morales V.  
Jorge A. Villalobos S.**

**Grupo DFAC  
Diseño y Fabricación Asistidos por Computador  
Departamento de Ingeniería de Sistemas y Computación  
Universidad de los Andes  
Apartado Aéreo 4976 -Bogotá, Colombia  
e-mail: [jvillalo@andescol.bitnet](mailto:jvillalo@andescol.bitnet)**

**Resumen:** Se presenta una formalización de los conceptos usados para describir los elementos del mundo de operación de un robot. Dichos conceptos pueden ser usados en una presentación formal de un lenguaje implícito de programación, basado en especificación de planes de solución de tareas, en términos de acciones, las cuales son traducidas en operaciones de robot [8]. Además, se incorpora el concepto de rasgos de ensamblaje (features), como información adicional para facilitar el proceso de traducción de las tareas de ensamble, tal y como está en las propuestas de manejo de rasgos de [7], [2] y [10].

**Palabras clave:** robótica, programación de robots, lenguajes implícitos, rasgos, planes de solución.

## 1. Introducción

El problema de la programación de robots se ha convertido en elemento clave del proceso de automatización industrial, porque de la facilidad de dicha programación, depende el éxito de incorporarlos en el desarrollo industrial del país. Por esta razón, se ha venido investigando en esta área: desde el desarrollo de ambientes de programación de robots [11], hasta herramientas automáticas de programación [8].

En este artículo se presenta un marco formal para el desarrollo de este tipo de herramientas, siguiendo el esquema de programación asistida por el programador. Utiliza un esquema de descripción de planes de solución e información de interés para el proceso de programación (rasgos de ensamble). A partir de esta especificación, es posible diseñar un modelo de compilador para un lenguaje automático y presentar un paradigma de programación de robots [8]. La estructura de este artículo es la siguiente: inicialmente se presenta el concepto de rasgo de ensamble, luego se presentan los conceptos a través de los cuales se construye el modelo del mundo y finalmente, se define validez de los elementos.

## 2. Rasgos

Los rasgos surgen como una propuesta para enriquecer las herramientas que apoyan el proceso de diseño (CAD/CAM), de tal forma que sea más natural para el diseñador expresarse. Consiste en agregar información que caracterice los elementos del proceso, facilitando la expresión de las acciones propias de éste. Una primera propuesta formal se presenta en [7] [2] y una especificación formal de un sistema basado en rasgos en [10].

Los rasgos (features) son características de los objetos, adicionales a la geometría, para manejar información que facilite el proceso en el que está involucrado este objeto. Dicha información caracteriza clases de objetos por comportamiento, forma, etc., adquiriendo una identidad, de tal forma que permiten expresar la semántica del proceso a través de ellos y facilitan la labor de los usuarios finales. Dentro del proceso de ensamblaje de piezas metalmecánicas (rígidas), utilizando robots, los rasgos expresan información que facilita las operaciones necesarias del robot para resolver una tarea de ensamblaje, como agarrar un objeto por un punto estable, transportarlo, ensamblarlo con otros objetos, etc.

### 2.1. Clasificación de rasgos de ensamblaje

**Rasgo de forma:** Es un conjunto de superficies pertenecientes a un objeto o ensamble que definen una característica interesante para el proceso de ensamblaje. Estos rasgos se caracterizan con un identificador, un sistema de referencia que define la localización del rasgo sobre el objeto y un conjunto de identificadores de objetos con los que se puede relacionar el objeto a través de este rasgo.

**Rasgo de manipulación:** Característica importante de un objeto, útil para el manejo de éste utilizando un robot. Estos rasgos se caracterizan con un identificador y un sistema de referencia que define la localización del rasgo sobre el objeto. Un ejemplo es un punto de agarre estable de una pieza.

**Rasgo físico:** Característica física de un objeto o ensamble. Se usa para informar a los procesos de planificación de ensamblaje condiciones de estabilidad, centro de masa, etc. Estos rasgos se caracterizan con un identificador, un indicador del tipo de característica física y un valor asociado con esa característica.

**Rasgo funcional:** Describe la semántica de un objeto o ensamble. Se usa para verificar la validez de las acciones realizadas por el proceso de ensamblaje. Se caracterizan con un identificador, un verbo que expresa la acción que realiza el objeto y un conjunto de objetos con los que está asociado el objeto a través de dicha funcionalidad.

## 2.2. Especificación de rasgos

A continuación se especifica un conjunto de rasgos base, usados para la especificación del modelo del mundo y su validez. La caracterización de este conjunto de rasgos, en particular, está dada para robots manipuladores de hasta seis (6) grados de libertad, usando una pinza de dos dedos paralelos, sin tener en cuenta características físicas como peso, estabilidad, inercia, etc.

**Sección de ensamblaje:** Es un rasgo de forma que consiste en un conjunto de superficies, que entran en contacto con una sección de ensamblaje de otro objeto o ensamble, estableciendo una relación de unión entre ellos. A partir de esta relación de unión, los objetos se hacen un solo ensamble.

**Superficie de soporte:** Una superficie de soporte es un rasgo de manipulación que consiste en un conjunto de superficies especificadas a través de un sistema de referencia que sirve para apoyar el objeto sobre otra superficie, manteniéndose estable.

**Punto de agarre:** Un punto de agarre es un rasgo de manipulación que indica una posición estable para agarrar un objeto o ensamble usando un robot. La caracterización de un punto de agarre es un identificador, un sistema de referencia que define la localización del punto sobre el objeto y una apertura de la pinza.

## 3. Conceptos básicos

A continuación, se definen los conceptos básicos para la especificación de los elementos del mundo de operación de un robot. Sea INDEF un constante que evalúa indefinido para la especificación.

**Un robot:** Un robot industrial R es una secuencia de N cuerpos o eslabones rígidos (grados de libertad) en forma de cadena cinemática abierta simple, a través de N pares rotacionales o prismáticos. Está caracterizado por una descripción geométrica de cada eslabón y por una descripción cinemática. Se representa por una tupla de la forma:  $R = \langle \text{nombre}, N, G, C, DH, [ \min 1, \max 1 ], \dots, [ \min N, \max N ], P \rangle$ , donde N es el número de grados de libertad, G es la geometría, C es el modelo cinemático, DH son los parámetros de Denavit-Hartenberg,  $[ \min 1, \max 1 ], \dots, [ \min N, \max N ]$  son los valores articulares máximos y mínimos para cada articulación y P es la pinza o herramienta que usa el robot.

**Variable de estado de un robot R** es el conjunto de valores que definen un estado único del robot en un momento dado, en términos de su posición y de si lleva un objeto en la pinza. Se representa por una tupla de la forma  $VER = \langle R, \emptyset 1, \dots, \emptyset N, \text{Apertura}, \text{Obj} \rangle$ , donde R es el robot,  $\emptyset 1, \dots, \emptyset N$  son los valores articulares del robot, Apertura define la apertura de la pinza y Obj es el objeto (si lleva) en la pinza.

**Un obstáculo Obt** del mundo es un elemento geométrico definido con respecto a un sistema de referencia local, más un conjunto de rasgos superficie de soporte. Los obstáculos se usan para describir el mundo de trabajo donde está el robot y son objetos que no pueden ser manipulados por el robot, únicamente pueden ser usados como elementos de apoyo para resolver las tareas. Un obstáculo se representa por una tupla de la forma  $Obt = \langle \text{nombre}, G, \{ R_1, \dots, R_k \} \rangle$ . El estado de un obstáculo es la transformación que define la ubicación del obstáculo respecto a un sistema de referencia de base. Se representa a través de una variable de estado de la forma  $VEObt = \langle Obt, T \rangle$ .

**Un objeto O** del mundo está definido por su geometría (con respecto a un sistema de referencia local) y un conjunto de rasgos. Se representa por una tupla de la forma  $O = \langle \text{nombre}, G, \{ R_1, \dots, R_k \} \rangle$ . El estado de un objeto es la transformación que define la ubicación del objeto respecto a un sistema de referencia base y el ensamble al que pertenece (si hay). Se representa a través de una variable de estado de la forma  $VEO = \langle O, T, \text{listaEnsamble} \rangle$ . Cuando la localización de un objeto no está definida, se asocia en su variable de estado la transformación indefinida.  $VEO = \langle O, INDEF, \text{listaEnsamble} \rangle$ . Igualmente, cuando no está definido el ensamble al que pertenece un objeto, se asocia al objeto el ensamble indefinido.  $VEO = \langle O, T, INDEF \rangle$ .

**Un ensamble E** del mundo es una relación definida para un conjunto de objetos, asociados explícitamente bajo un nombre NE, en el cual un conjunto de superficies de cada objeto en el ensamble entra en contacto con superficies de otro objeto o ensamble en NE, de tal forma que se comporten en forma solidaria. Cuenta con un sistema de referencia local, una transformación que lo ubica en el sistema de referencia del mundo y por un conjunto de rasgos del ensamble. Está definido por el conjunto de objetos que lo constituye más un conjunto de rasgos. Se representa por una tupla de la forma  $E = \langle NE, O_1, \dots, O_n, \{ R_1, \dots, R_k \} \rangle$ . El estado de un ensamble es la transformación que define la ubicación del ensamble respecto a un sistema de referencia base (el mundo), durante una tarea dada. Se

representa a través de una variable de estado definido por una tupla de la forma  $VEE = \langle E, T \rangle$ . Donde T es una transformación que define la localización de cada uno de los objetos en el ensamble de la forma Localización (  $O_i$  ) = T.T*O*<sub>i</sub> Para todo  $O_i$  en E. Cuando la localización del ensamble no está definida, no no lo está para ninguno de los objetos en el ensamble.  $VEE = \langle E, INDEF \rangle \rightarrow$  Localización (  $O_i$  ) = INDEF Para todo  $O_i$  en E

Un escenario ESC es un sistema de referencia SR y un conjunto de obstáculos que describen los elementos estáticos en el ambiente de trabajo de un robot. Se representa como una tupla de la forma  $ESC = \langle SR, \{ Obt1, \dots, Obt_n \} \rangle$ . El estado de un escenario se describe a través de un conjunto de variables de estado de obstáculos, que definen la localización de los obstáculos con respecto al sistema de referencia SR. La variable de estado de un escenario es una tupla de la forma  $VEESC = \langle ESC, \{ VEObt1, \dots, VEObt_n \} \rangle$ .

El mundo M es un conjunto de objetos y un conjunto de ensamblajes dado un escenario, que describe el ambiente donde se resolverá una tarea con los objetos descritos. El mundo es usado para crear un modelo de las tareas de ensamblaje que se vayan a resolver y se representa como una tupla de la forma  $M = \langle ESC, \{ O_1, \dots, O_n \}, \{ E_1, \dots, E_m \} \rangle$

La variable de estado del mundo VEM es un conjunto de variables de estado de los objetos y de los ensamblajes, dada una variable de estado de un escenario. El escenario tiene una variable de estado VEESC con un valor constante para todos los valores de las variables de estado de los objetos y ensamblajes en el mundo.

Una escena E es el estado del modelo del mundo M en un instante de tiempo dado. Equivale a una valuación de la variable de estado del mundo con respecto al tiempo t, dado en alguna unidad:  $E = VEM / t$

Un bosquejo de escena BE es una escena en la cual se encuentran indefinidas algunas características de los objetos en el mundo, tales como su localización o la definición de ensamblajes.

Una tarea atómica de ensamblaje TA es una transformación del mundo M donde solo se modifica la localización de un objeto y ninguna o varias relaciones entre ellos. Está definida por dos escenas del mundo:  $TA = \{ E_1, E_2 \}$ . TA puede ser cualquier transformación del mundo obtenida por una de las siguientes acciones de ensamblaje: agarrar, soltar, insertar, mover, localizar, girar, unir o desunir.

Una tarea de ensamblaje T es una transformación del mundo M en la cual solo se modifica la localización de los objetos y las relaciones entre ellos, no su geometría ni sus propiedades. Está definida por dos escenas del mundo que describen una situación inicial y una situación final. Estas escenas inicial y final se conocen como Precondición y Postcondición de la tarea:  $T = \{ \text{Precondición ( PRE )}, \text{Postcondición ( POST )} \}$

Un plan de solución P para una tarea T especificada por ( PRE, POST ) es una secuencia de tareas atómicas TA1 ... TAN tales que la situación inicial de TA1 es PRE, la situación final de TAN es POST y la situación final de TAI es igual a la situación inicial de TAI+1. Si PRE es un estado válido en el mundo real y POST es un estado válido en el mundo real, puede existir un plan P, tal que se cumple ( PRE ) P ( POST ), aunque, P no necesariamente es ejecutable para todo robot R.

El programa de ensamblaje PE correspondiente a un plan P, está definido por la secuencia de acciones de ensamblaje A1 ... An asociadas con cada una de las tareas atómicas en el plan P.

Un borrador de solución B para una tarea T especificada por ( PRE, POST ) es una secuencia de tareas T1 ... Tn especificadas de tal forma que no definen completamente la secuencia de tareas atómicas por realizar o es una secuencia de tareas atómicas definidas mediante bosquejos de escena o a través de descripciones incompletas de acciones de ensamblaje o es una combinación de tareas atómicas con bosquejos de tareas y tareas complejas.

Un programa del robot PR es una secuencia de operaciones de movimiento sobre un robot R expresada en un lenguaje explícito, que efectúa la tarea T definida por ( PRE, POST ) siguiendo un plan especificado en un programa de ensamblaje P.

**Programación implícita:** Dado un modelo del mundo M, un robot R y una tarea T que describe el ensamblaje que se desea desarrollar, se debe construir un plan P de solución y un programa de robot PR que haga las transformaciones definidas en P para resolver T. El problema de la validez de la tarea respecto al modelo del mundo M y al mundo real se trata en la siguiente sección.

Una heurística H es una función de probabilidad f, tal que  $f(x) = P$ , donde P es el valor de probabilidad asociado al evento x. La aplicación que se le da en este artículo es la de calificar las posibles soluciones, según un criterio dado por la función, para resolver problemas de planificación de trayectorias, de forma que, indique cual es la opción con mayor probabilidad.

Un planificador PI es una función que dada la tarea por ejecutar f, una heurística H para calificar las soluciones de f y un entero i, cumple que  $PI ( f, H, i ) = S$ , donde S es la i-ésima solución de f según H.

#### 4. Corrección y verificación de planes

El problema de corrección y verificación de planes, se trata en forma distinta a la aplicable en programación clásica de computadores, por las razones expuestas en [8]. A continuación, se presenta el esquema que se debe seguir para construir escenarios y escenas correctas y válidas respecto a una escena del mundo real, de manera que, especificando

de esta forma las tareas (PRE, POST) y generando las escenas del plan siguiendo este esquema, se garantiza al menos consistencia en el modelo.

**Posición válida de un obstáculo:** Un obstáculo  $Obt$  está en una posición válida  $P$  respecto al sistema de referencia del escenario  $ESC$  si la posición se define con una de las siguientes reglas:

- $P$  está definida por  $T \neq INDEF$  y  $Obt$  no se relaciona con ningún otro obstáculo en  $ESC$ .
- $P$  está definida por  $T = Obt.SSi \cdot Obj.SSk$  y  $Obj$  está en posición válida.

**Definición de un escenario:** Un escenario  $ESC$  válido se define como una valuación de  $VEESC$  donde todas las posiciones de los obstáculos son válidas y se definen con respecto al sistema de referencia del escenario o con respecto a un obstáculo válido en el sistema de referencia.

**Posición válida de un objeto:** Una posición válida  $P$  de un objeto  $O$  es una posición del objeto definida para un escenario válido  $ESC$ , aplicando alguna de las siguientes reglas:

- $P$  está definida por  $T = O.SSi \cdot Obj.SSk$  y  $Obj$  pertenece a  $ESC$ .
- $P$  está definida por  $T = O.SSi \cdot Oj.SSk$  y  $Oj$  está en posición válida en  $ESC$ .
- $P$  está definida por  $T = O.SEi \cdot Oj.SEk$  y  $O$  está en posición válida en  $ESC$  y  $O$  se relaciona con  $Oj$  a través de  $SEi$  y  $Oj$  se relaciona con  $O$  a través de  $SEk$ .

Y no quedar en colisión con ningún elemento en posición válida (del escenario, objeto o ensamble).

**Posición en conflicto:** Un objeto  $O$  u obstáculo  $Obt$  está en posición en conflicto en una escena si  $O$  u  $Obt$  está en ubicado en posición válida respecto al escenario, pero  $O$  u  $Obt$  queda en colisión con otro elemento  $Ej$  en posición válida y ubicado antes que el objeto  $O$ .

**Definición de una bosquejo de escena:** Un bosquejo de escena válido  $BE$  se define como un bosquejo de escena donde el escenario es un escenario válido  $ESC$  y el conjunto de objetos y ensambles en el bosquejo de escena, están ubicados en: posiciones válidas respecto al escenario, posiciones válidas respecto a objetos válidos respecto al escenario, la posición  $INDEF$ .

**Definición de una escena:** Una escena válida  $E$  se define como un bosquejo de escena donde ningún objeto o ensamble está ubicado en la posición  $INDEF$ .

**Definición de un plan:** Un plan  $P = TA1, \dots, TAN$  es válido si:

- Para todo  $TAi, TAj, ESCi = ESCj$ .

- Para todo  $TA_i = \{ E1, E2 \}$ , E1 y E2 son escenas válidas respecto al escenario.
- Entre  $TA_i$  y  $TA_{i+1}$  únicamente cambia de posición un objeto o se genera o destruye un ensamble.

**Definición de un borrador de plan:** Un borrador de plan es válido si cumple las dos primeras reglas de la definición de un plan.

**Refinamiento de un bosquejo de escena:** Un bosquejo de escena se refina cuando un objeto o ensamble definido en la posición INDEF, se redefine en una posición respecto a un obstáculo, objeto o ensamble en posición válida en el bosquejo de escena. Sin embargo, a pesar de seguir siendo un bosquejo de escena válido o de haberse convertido en escena válida, la validez del bosquejo depende de su validez en el contexto en el que se halle, en este caso un borrado de plan.

Para que el bosquejo de escena siga siendo válido en el borrador, la redefinición debe cumplir que: Sea  $BP = T1, \dots, Tn$ . Sin pérdida de generalidad, podemos decir que hay un tarea  $T_i$  tal que  $T_i = \{ E_j, BE \}$ , y que  $T_{i+1}$  cumple que  $T_{i+1} = \{ BE, E_k \}$ . Sea  $BE = VEM / t$ , donde  $VEOr = \langle Or, INDEF, listaEnsamble \rangle$ . Si se redefine la posición de Or, en una posición válida P, en el bosquejo diferente de la posición INDEF, debe cumplir que:

- Si de  $T_{i+1}, \dots, T_n$  la posición de Or no cambia (es INDEF), entonces la nueva posición de Or no debe quedar en posición en conflicto con ningún objeto definido entre  $T_{i+1}$  y  $T_n$ .
- Si en un  $T_s, s > i$ , Or es definido en una posición P1, debe aparecer un TA entre  $T_{s-1}$  y  $T_s, T_{s'}$  donde Or pase de la posición P a P1 y entre  $T_{i+1}$  y  $T_{s'}$  se cumple la regla anterior.

## 5. Conclusiones

La especificación del modelo está construida con base en tres rasgos básicos, secciones de ensamble, superficies de soporte y puntos de agarre, que componen una base para las operaciones permitidas, (agarrar - llevar-dejar), pero el modelo es extensible para usar más rasgos que faciliten determinados procesos.

El trabajo de formalización en el área de programación de robots, da piso al desarrollo de herramientas automáticas que faciliten el proceso de programación y permite, en un futuro, incorporar especialistas en estrategias de solución de planes.

El modelo de programación basado en planes, ofrece una gran flexibilidad, no solo para implantarlo sino para extenderlo al manejo de sistema sensorial y replanificación.

## Bibliografía

[1] Basañez, L., Operation Specialists for Automatic Programming and Monitoring of Robotic Assembly Cells, Journal of Robotics and Computer Integrated Manufacturing, 1989.

- [2] De la Rosa, F., Una Propuesta de Diseño Asistido por Computador de Piezas Metalmecánicas Representados por Rasgos, Universidad de Los Andes, Tesis de Magister, Junio 1991.
- [3] Laugier, C., Raisonement Geometrique et Methodes de Decision en Robotique. Aplicacion a la Programmation Automatique des Robots, Tesis de doctorado, Institut National Polytechnique de Grenoble, diciembre 1987.
- [4] Lieberman, L., AUTOPASS: An Automatic Programming System for Computer Controlled Mechanical Assembly, IBM Journal of Research and Development, Vol. 21, No. 4, Julio 1977.
- [5] Lozano-Perez, T., Lama: A language for Automatic Mechanical Assembly, MIT.
- [6] Lozano-Perez, T., The design of a Mechanical Assembly System, MIT, 1976.
- [7] Manthila, M., A Modeling System for Top-Down Design of Assembled Products, IBM Journal Res. Develop, Vol. 34, No. 5, septiembre 1990.
- [8] Morales, J., Un ambiente de programación de robots con capacidad de planificación de tareas, Tesis de Magister, Universidad de Los Andes, Informe Interno, 1991.
- [9] Pertin, J., Modelisation du Raisonement Geometrique pour la Programmation des Robots, tesis de Doctorado, Institut National Polytechnique de Grenoble, 1986.
- [10] Ramos, O., OF, tesis de Magister, Universidad de los Andes, 1991.
- [11] Villegas, E., Ambientes de Programación de Robots, Tesis de Magister, Universidad de Los Andes, 1991.