

Planificación en agentes BDI

Diego García
drg@cs.uns.edu.ar

Guillermo Simari
grs@cs.uns.edu.ar

Alejandro García
ajg@cs.uns.edu.ar

Laboratorio de Investigación y Desarrollo en Inteligencia Artificial (LIDIA)
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur.

Av. Alem 1253, (8000) Bahía Blanca, Argentina
Tel: (0291) 459-5135 / Fax: (0291) 459-5136

1 Introducción

Una de las metas de la Inteligencia Artificial es la construcción de agentes autónomos racionales, capaces de actuar en ambientes complejos y dinámicos. Una de las arquitecturas más difundidas y estudiadas para especificar este tipo de agentes, es la arquitectura BDI [1]. Este modelo tiene sus raíces en la tradición filosófica de entender el razonamiento práctico, el cual involucra dos procesos importantes: *deliberación*, decidir que metas se desean realizar, y *razonamiento de medios-fines*, como obtener dichas metas.

Si bien la habilidad de hacer y ejecutar planes es un ingrediente esencial para un agente BDI, la mayoría de las implementaciones conocidas de agentes BDI usan bibliotecas de planes definidas para el agente previamente a su ejecución [2, 3]. El uso de bibliotecas de planes es más eficiente que planificar en tiempo de ejecución, y permite manejar la complejidad del razonamiento de medios-fines. Sin embargo restringe las habilidades del agente a las situaciones previstas por su diseñador. Planificar en tiempo de ejecución permitiría al agente enfrentar una variedad de situaciones más amplia.

Por otra parte los problemas de planificación que enfrenta un agente de este tipo, contrastan de manera significativa con los tipos de problemas que son resueltos por la tecnología de planificación actual. Esto se debe principalmente a que los sistemas de planificación tradicionales asumen simplificaciones sobre los problemas de planificación para disminuir su complejidad.

El propósito de esta línea de investigación es la incorporación y adaptación de técnicas de planificación al modelo BDI, con el objetivo de mejorar el proceso de razonamiento de medios-fines para aumentar la autonomía del agente.

2 Breve Descripción de la arquitectura BDI

El modelo BDI (Belief-Desire-Intentions) es posiblemente una de las arquitecturas más prometedoras para el desarrollo de agentes inteligentes, y se ha convertido en uno de los modelos más difundidos y estudiados de la literatura. Un agente de arquitectura BDI posee un conjunto B de creencias acerca de su entorno, un conjunto D de deseos y un conjunto I de intenciones.

Financiado parcialmente por la Agencia Nacional de Promoción Científica y Tecnológica (PICT 2002 Nro 13096) y por SeCyT Universidad Nacional del Sur (subsidio: 24/ZN09).

La arquitectura BDI tiene sus raíces en el denominado Razonamiento Práctico [1]. Este tipo de razonamiento puede describirse como el proceso de decidir qué acción realizar para alcanzar las metas, e involucra dos importantes procesos: (a) Deliberación, esto es decidir qué metas se quieren alcanzar, y (b) Razonamiento de medios-fines (means-ends): decidir cómo se alcanzarán las metas.

Las intenciones de un agente son un subconjunto de las alternativas que éste posee para alcanzar sus metas. Una de las características más importantes de las intenciones es su rol motivador, i.e. provocan acciones. Una vez adoptada una intención, ésta afectará el razonamiento práctico que considere en el futuro. Las intenciones tienen la propiedad de persistencia, es decir para que estas resulten útiles, el agente no deberá abandonar sus intenciones, sino que las mantendrá hasta que las cumpla, o bien sea evidente que no podrá cumplirlas, o bien las razones que tuvo para adoptarla dejaron de ser válidas. Sus intenciones están relacionadas con sus creencias acerca del futuro.

El modelo básico de la arquitectura BDI (ver Figura 1) propone el siguiente esquema de comportamiento para los agentes:

1. Percibir los cambios del entorno en el que se desenvuelve.
2. Revisar sus creencias acerca del mundo en base a su percepción.
3. Razonar acerca de sus intenciones a fin de reconsiderarlas en caso que sea necesario.
4. Seleccionar una acción a seguir, razonando sobre sus creencias y sus intenciones.
5. Ejecutar la acción seleccionada, la cual producirá cambios en el entorno, y volver al paso 1 a fin de percibir los cambios que se han producido.

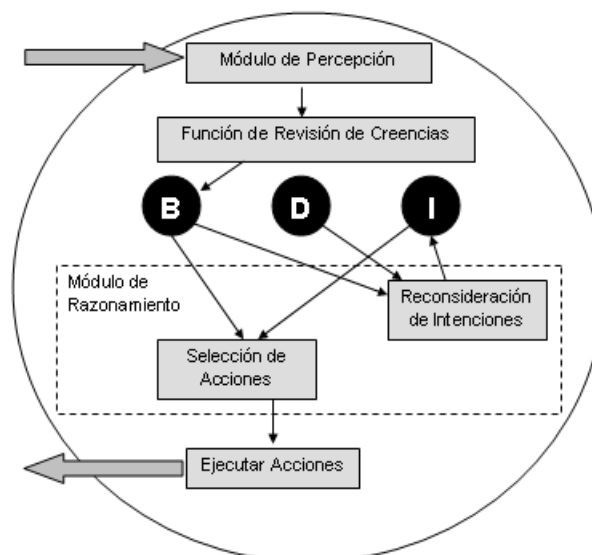


Figura 1: Esquema de la arquitectura BDI básica

3 Línea de investigación actual

Como mencionamos anteriormente, si bien la habilidad de generar y ejecutar planes es un ingrediente esencial para un agente BDI, la mayoría de las implementaciones conocidas de agentes BDI usan bibliotecas de planes, los cuales son simplemente seleccionados por el agente y luego instanciados a la situación particular.

Por ejemplo en [2] un agente consiste de una base de datos con las creencias actuales, un conjunto de las metas a ser realizadas, un conjunto de planes llamado *Knowledge Areas* (KAs) y una estructura de intenciones que contiene aquellos planes que fueron elegidos para ejecutarse. Cada KA consiste de un *cuero* que describe los pasos del procedimiento, y una *condición de invocación* que especifica bajo que situaciones es aplicable. El *cuero* puede ser visto como un plan o en esquema de plan.

Otro ejemplo donde se utilizan bibliotecas de planes es el lenguaje AgentSpeak(L) [3]. Se trata de un lenguaje de programación basado en un lenguaje de primer orden con acciones y eventos, diseñado para caracterizar agentes bajo la arquitectura BDI. Este lenguaje permite definir un conjunto de planes pre-compilados que el agente puede usar en ejecución, y no requieren ningún tipo de planificación por parte del mismo. Un plan especifica los medios por los cuales un agente logrará un fin y consiste de una *cabeza* y un *cuero*. La *cabeza* del plan consiste de un *evento disparador* (denotado +!E) y un *contexto* (separado por ":",") que especifica que creencias debe tener el agente para que el plan pueda ser disparado. El *cuero* del plan es una secuencia de metas (que el agente debe lograr o testear) y acciones (que el agente debe ejecutar). Por ejemplo un plan para que un robot cambie de posición tendría la siguiente descripción en el lenguaje AgentSpeak(L):

```
+!location(robot,X): location(robot,X) <- true.  
  
+!location(robot,X): location(robot,Y) &  
                      (not (X=Y)) &  
                      adjacent(Y,Z)  
                      <- move(Y,Z); +!location(robot,X).
```

El uso de bibliotecas de planes resulta mas eficiente que planificar en tiempo de ejecución y permite manejar la complejidad del razonamiento de medios-fines, sobre todo en ambientes dinámicos donde existen fuertes restricciones de tiempo. Sin embargo, restringe las habilidades del agente únicamente a las situaciones previstas en la biblioteca de planes.

Planificar en tiempo de ejecución permite al agente enfrentar un variedad de situaciones más amplia y aumentar su autonomía. En [4] se presenta una extensión al modelo BDI que incorpora planificación proposicional y utiliza Graphplan [5] como algoritmo de planificación. Si bien es posible incorporar métodos de planificación a la arquitectura DBI, la tecnología de planificación actual resulta inadecuada [8] para resolver los tipos de problemas que enfrenta un agente BDI capaz de actuar en un ambiente complejo y dinámico. Esto se debe a que los métodos tradicionales de planificación asumen simplificaciones sobre los problemas planificación, que resultan adecuadas para otros dominios y disminuyen la complejidad de los problemas. Las simplificaciones mas importantes son:

- El planificador tiene todo el conocimiento relevante que necesita en el momento que comienza a planificar.
- Nada importante ocurre en el mundo, excepto como resultado de sus acciones. Se ve al planificador como un modulo separado, en lugar de estar integrado a la arquitectura cognitiva del agente

En un ambiente dinámico estas suposiciones pueden fallar debido a que [8]:

- Las creencias del agente son falibles, a medida que adquiere mas información sus creencias pueden cambiar.
- A medida que el mundo cambia, las creencias del agente deben actualizarse.
- El conocimiento del agente acerca del mundo, por mas preciso que sea, será incompleto

Estos sugiere el desarrollo de nuevas técnicas de planificación, o la modificación y adaptación de las técnicas tradicionales.

La idea de esta linea de investigación es desarrollar un modelo híbrido, que conserve una biblioteca de planes para mantener la eficiencia en situaciones donde el agente debe responder rápidamente, y disponer de un planificador adecuado para resolver situaciones no contempladas por los planes de la biblioteca. De esta manera, ante una situación particular, si existe un plan aplicable en la biblioteca el agente podrá utilizarlo directamente. De lo contrario, el agente generaría un plan, lo utilizaría y si tuvo éxito lo almacenaría en la biblioteca para situaciones futuras. Este esquema sugiere de alguna manera, que el agente almacena su experiencia de situaciones previas, para manejar situaciones en el futuro. En trabajos anteriores [6] se desarrollo una extensión al algoritmo de Graphplan que durante la resolución de un problema de planificación genera conocimiento automáticamente (experiencia) que puede ser utilizado para resolver nuevos problemas del mismo dominio de planificación. En esta investigación se propone utilizar una variante de Graphplan como planificador de propósito general. La biblioteca de planes tendrían una estructura particular donde se mantendría el grafo de planificación asociado a un plan, junto con el conocimiento adquirido en la generación del mismo.

4 Conclusiones

El uso de bibliotecas de planes resulta mas eficiente que planificar en tiempo de ejecución y permite manejar la complejidad del razonamiento de medios-fines, sobre todo en ambientes dinámicos donde existen fuertes restricciones de tiempo. Sin embargo, restringe las habilidades del agente únicamente a las situaciones previstas en la biblioteca de planes. Planificar en tiempo de ejecución permite al agente enfrentar un variedad de situaciones más amplia y aumentar su autonomía. Sin embargo, los planificadores de propósito general actuales no resultan adecuados para un agente con conocimiento incompleto que se desenvuelve en un entorno cambiante.

En esta investigación se propone utilizar una variante de Graphplan como planificador de propósito general para la arquitectura BDI. El agente BDI dispondría de una biblioteca de planes, si existe un plan aplicable en la biblioteca el agente podrá utilizarlo directamente. De lo contrario, el agente generaría un plan, lo utilizaría y si tuvo éxito lo almacenaría en la biblioteca para situaciones futuras.

Referencias

- [1] M. E. Bratman, D.J.Israel, M.E.Pollack. *Plans and Resource-Bounded Practical Reasoning*. Computational Intelligence, 4: pp. 349-355, 1988.
- [2] Michael P. Georgeff, Francois Félix Ingrand. *Decision-Making in an Embedded Reasoning System*, in Proceedings of the Eleventh International Joint Conference on Artificial Intelligence, Detroit (Michigan), August 1989.
- [3] A. S. Rao, M. P. Georgeff. *BDI Agents: From Theory to Practice*, in Proceedings of the First International Conference on Multi-Agent Systems (ICMAS-95), San Francisco, USA, pp. 312-319, 1995.
- [4] Felipe Rech Meneguzzi, Avelino Francisco Zorzo, Michael Da Costa Móra. *Propositional Planning In BDI Agents*, 2004 ACM Symposium on Applied Computing.
- [5] A. L. Blum and M. L. Furst. *Fast Planning through planning graph analysis* Journal of Artificial Intelligence, 90 (1-2):281-300, 1997.
- [6] Diego García y Alejandro García *Una Extensión de Graphplan con Aprendizaje y Argumentación para la Elección de Acciones*. IX Congreso Argentino de Ciencias de la Computación (CACIC 2003). Universidad Nacional de La Plata, La Plata, Argentina, págs 800-811, Octubre de 2003.
- [7] R. Kambhampati, E. Lambrecht and E. Parker. *Understanding and Extending Graphplan* In Proc. 4th European Conference on Planning. September 1997.
- [8] John L.Pollock. *Defeasible Planning. Integrating Planning, Scheduling, and Execution in Dynamic and Uncertain Environments* AIPS Workshop. Ralph Bergmann and Alexander Kott, Cochairs. 1998, Madison Wisconsin.
- [9] Daniel S. Weld. *Recent advances in AI planning*. *AI Magazine*, 20(2):93–123, 1999.