

# Planning en agentes inteligentes

Luis S. Berdun

Instituto ISISTAN, Facultad de Ciencias Exactas,  
Universidad Nacional del Centro de la Pcia. de Buenos Aires  
Campus Paraje Arroyo Seco – (7000) – Tandil- Bs. As. -Argentina  
Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET)  
email: [lberdun@exa.unicen.edu.ar](mailto:lberdun@exa.unicen.edu.ar) TE:(02293) 440363 int 28

## Resumen

Los agentes inteligentes necesitan satisfacer sus objetivos, para esto una de las técnicas más utilizadas es planning. El problema es que los algoritmos de planning no fueron pensados especialmente para interactuar con agentes, sino para que estos aprovechen sus virtudes a fin de armar un plan que logre satisfacer sus objetivos. La mayoría de los agentes inteligentes poseen información extra sobre el ambiente en el que se encuentra, esta información conforma el estado mental del agente (creencias, preferencias, objetivos, etc.). La idea principal del proyecto es utilizar internamente en la construcción del plan la información del estado mental del agente.

## Introducción

Los agentes inteligentes de software son componentes computacionales que actúan autónomamente para alcanzar un conjunto de objetivos. Para satisfacer estos objetivos, uno de los algoritmos más utilizados es el de planning. Un algoritmo de planning construye un plan de acción a partir de un estado inicial de su mundo, un estado final que desea alcanzar y un conjunto de acciones que le podrían permitir llegar al objetivo.

Los algoritmos de planning son actualmente utilizados por componentes que representan agentes inteligentes en forma aislada, invocando estos algoritmos sólo enviando los tradicionales datos de estado inicial, estado final y acciones. Sin embargo, los agentes necesitan que estos algoritmos contemplen su estado mental. En el estado mental de los agentes se representan aquellas actitudes mentales que le permiten decidir entre varios caminos a seguir sin que estos estén explícitamente codificados en su implementación. Así, los objetivos, las preferencias, los compromisos, las intenciones, entre otras actitudes mentales, son representadas en el estado mental de un agente.

Esta necesidad de los agentes se evidencia en el diseño de agentes de interfaz, por ejemplo, cuando preferencias y compromisos guían las definiciones de que planes resultan viables o no, y que planes permiten a sus agentes obtener más ventajas que con otros. Para alcanzar esta funcionalidad con los algoritmos de planning tradicionales, es necesario realizar un análisis de cada plan posible filtrando los planes que contradicen fuertemente preferencias y compromisos y luego generando un ranking para maximizar ventajas. Lamentablemente, el tiempo consumido en este proceso es demasiado grande para hacerlo viable en aplicaciones de agentes de interfaz ya que éstos requieren interacción constante con un usuario humano a partir de estos resultados.

## Planning

Tomando como algoritmo de experimentación un algoritmo que retorna un plan con orden parcial, el algoritmo UCPOP [Weld 94], considera que este recibe como datos iniciales tres elementos como se puede observar en el esquema de algoritmo presentado a continuación. El primer dato es un plan nulo que contiene el estado actual, el segundo es una agenda de objetivos y el tercero es un conjunto de acciones con sus precondiciones y efectos.

En el armado del plan, tres elementos son definidos,  $\langle A, O, L \rangle$  en el algoritmo. A es el conjunto de acciones que forman el plan, O es el conjunto de restricciones de orden y L es el conjunto de enlaces causales definidos durante la generación del plan. Un enlace causal especifica la razón por la cual una acción es ingresada a un plan, o sea, que efecto de una acción es condición necesaria para la ejecución de otra.

POP(  $\langle A, O, L \rangle$ , agenda, X )

1. *Terminación*: Si la agenda de objetivos está vacía, retorna  $\langle A, O, L \rangle$ .
2. *Selección del objetivo*: Sea  $\langle Q, A_{need} \rangle$  un par de la agenda (por definición  $A_{need} \in A$  y Q es un elemento de la precondición de  $A_{need}$ ).
3. *Selección de la acción*: Sea  $A_{add} = \text{choose}$  de una acción que alcanza Q (con una nueva acción de X, o una acción existente en A, la cual puede ser consistentemente ordenada como anterior a  $A_{need}$ ). Si tal acción no existe, se retorna fracaso. Sea  $L' = L \cup \{A_{add} -Q \rightarrow A_{need}\}$ , y sea  $O' = O \cup \{A_{add} < A_{need}\}$ . Se  $A_{add}$  es una nueva instancia, entonces  $A' = A \cup \{A_{add}\}$  y  $O' = O \cup \{A_0 < A_{add} < A_s\}$  (caso contrario  $A' = A$ ).
4. *Cambios en el conjunto objetivo*: Sea agenda' = agenda -  $\{\langle Q, A_{need} \rangle\}$ . Si  $A_{add}$  es una nueva instancia, entonces para cada conjunto  $Q_i$  de sus precondiciones, agregar  $\langle Q_i, A_{add} \rangle$  a agenda'.
5. *Protección de enlaces causales*: Para cada acción  $A_t$  que puede amenazar un enlace causal  $A_p -R \rightarrow A_c \in \text{choose } L'$  que es una restricción de orden consistente, hacer
  - a. Agregar  $A_t < A_p$  a  $O'$ , o
  - b. Agregar  $A_c < A_t$  a  $O'$  (promoción).
6. *Invocación recursiva*: POP( $\langle A', O', L' \rangle$ , agenda', X).

El esquema de algoritmo anterior expone los principales análisis realizados sobre los datos enviados para la generación de un plan de acción. En éste se remarca una invocación a una función choose, ya que es un punto importante que hace que el plan resultante, si buscamos sólo uno cualquiera, sea más eficiente. Es el único punto que se sugiere especializar en los diferentes dominios en los cuales se puede aplicar un algoritmo de planning.

### Estados mentales

El estado mental de un agente es el sub-componente computacional que tiene la responsabilidad de registrar no sólo el conocimiento que tiene sino también registrará otras actitudes mentales que permitirán procesar la toma de decisiones en forma autónoma del agente. Entre estas actitudes mentales, los objetivos son muy relevantes ya que son los que guían el comportamiento del agente. No menos importantes son las intenciones, que representan lo que el agente quiere. A partir de las intenciones, el agente decide intentar alcanzar alguna de ellas, momento en el cual se convierten en objetivos. Otras actitudes mentales influyen la decisión de que intención se convierte en objetivo y cómo estos objetivos son alcanzados. Entre estas últimas, se puede mencionar las creencias y las obligaciones.

Por ejemplo, `goal(box(182,londres))` especifica que el agente quiere que la caja 182 esté en Londres. Para ellos existen varios planes de acción posible que un agente puede calcular con un algoritmo de `planning`. Pero, si se considera que el agente tiene en su estado mental `preference(visit(paris),9)` que especifica que el agente tiene una preferencia alta (nueve) de visitar Paris, un plan que pase por Paris será de mayor preferencia que uno que no lo haga. Más aún, si visitar Paris no es una preferencia sino un objetivo `goal(visit(paris))` la importancia de esta restricción es aún mayor ya que cumpliría otro de sus objetivos, aunque no sea parte del planteo inicial.

Varios marcos formales se han construido para representar formalmente estas actitudes y las relaciones entre ellas [Cohen 90] [Singh 99] [Wooldridge 00]. Lamentablemente, no se han podido definir caminos de materialización prácticas de estos formalismos. Sin embargo, estas formalizaciones pueden ser utilizadas para enriquecer representaciones lógicas respetando estas definiciones de actitudes mentales. Particularmente, utilizamos módulos lógicos.

Los módulos lógicos permiten trabajar con cláusulas lógicas a nivel de programación, particularmente, el lenguaje `JavaLog` [Zunino 01] permite utilizar módulos lógicos y combinarlos con las operaciones definidas en [O’Keefe 85].

Por ejemplo, la figura 1 expone un ejemplo de un agente que define en tres módulos lógicos preferencias que utilizará en forma aislada o combinada en diferentes contextos.

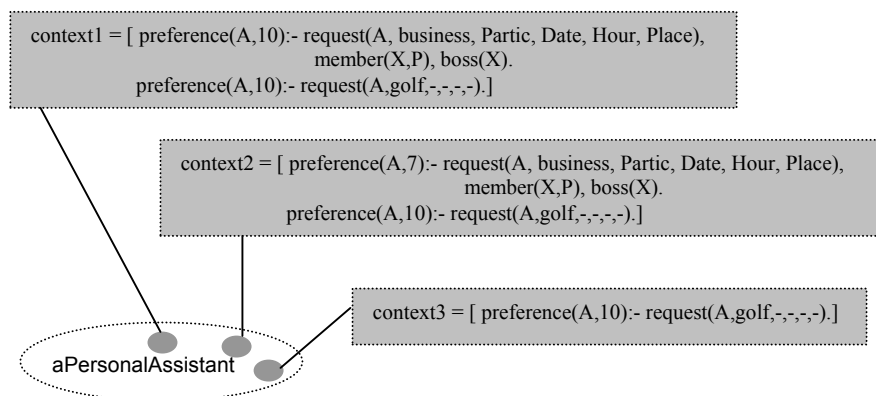


Figura 1: Módulos lógicos en un agente.

Estos módulos lógicos permiten combinar a través de un conjunto definido de operaciones cláusulas lógicas tipo Prolog. Enriqueciendo estos módulos con actitudes mentales se permitirá materializar la propuesta que se expone a continuación.

### Solución Propuestas

Los agentes inteligentes que en la toma de decisiones generan planes en tiempo de ejecución requieren que los planes que generan respeten su estado mental. Así, si contamos con dos planes posibles para alcanzar un objetivo pero uno de ellos viola nuestras preferencias, se tomará el restante para su ejecución. Una simple opción es la de generar todos los planes posibles y luego procesarlos considerando cada una de las actitudes mentales del estado mental del agente. Lamentablemente, los tiempo de ejecución hacen que esta solución no sea viable en agentes que conversan con usuarios humanos [Berdun 02].

Para resolver este problema, se propone el desarrollo de un algoritmo de planning específico para agentes, el cual considere internamente las actitudes mentales durante la generación de los planes. Para esto es necesario establecer puntos de interacción con el estado mental del agente, de manera tal que el agente se transforme en la guía del algoritmo, con esto se logra filtrar los planes a medida que se generan, evitando llegar a un plan que no este de acuerdo con las preferencias del agente. Adicionalmente, y en base a las preferencias y objetivos del agente, se busca armar una solución que se aproxime a la óptima, pero con un costo computacional mucho menor que el de recorrer todo el espacio de soluciones en búsqueda de la mejor.

Los trabajos iniciales se desarrollan sobre el algoritmo Ucpop. En base a este, se estudiaron posibles puntos de interacción con el agente, así como también que es lo que el agente necesita del algoritmo para poder trabajar con su estado mental. Esto permitió arribar a soluciones diferentes de las que se conseguían en una ejecución simple del algoritmo Ucpop. Las soluciones que se consiguen, de acuerdo con el estado mental del agente, se encuentran de acuerdo con las preferencias, objetivos y restricciones del agente.

### **Conclusiones**

El avance del proyecto permite hoy que un agente guíe la construcción del plan. Esto ha permitido obtener soluciones que varían de acuerdo al estado mental del agente. Particularmente se trabajo especialmente sobre el algoritmo Ucpop, logrando modificaciones que le permiten al agente manipular la creación del plan, y poder aplicar sus preferencias y obligaciones a medida que se gesta el plan, lo cual significa una mejoría temporal substancial respecto al trabajo realizado en [Berdun 02]. Paralelamente se construyó una herramienta que permite realizar seguimiento gráfico del plan en desarrollo y del estado mental del agente.

### **Referencias**

- [Berdun 02] Berdun L. Agentes filtrando planes según su estado mental. Tesis de grado de la carrera de Ingeniería en Sistemas de la Universidad Nacional del Centro de la Pcia. de Bs. As., 2002.
- [Cohen 90] Cohen P., Levesque H. Intention is Choice with Commitment. Artificial Intelligence, vol.42, no.2/3, march 1990.
- [O'Keefe 85] O'Keefe. Towards an Algebra for Constructing Logic Programs. In: J. Cohen and J. Conery (eds), Proceedings of the IEEE Symposium on Logic Programming, IEEE Computer Society Press, pp. 152-160, 1985.
- [Singh 99] Singh M., Rao A., Georgeff M. Formal methods in DAI: Logic-based representation and reasoning. In: Multiagent Systems, editor: Weiss G. MIT Press, 1999.
- [Weld 94] Weld Daniel S. A Introduction to least commitment planning. AI Magazine Summer / fall 1994
- [Wooldridge 00] Wooldridge M. Reasoning about Rational Agents. MIT Press, 2000.
- [Zunino 01] Zunino A., Berdún L., Amandi A. JavaLog: Un lenguaje para la programación de agentes. Inteligencia Artificial, n.13. pp. 94-99. 2001.