Anexo I: Invocación del planificador desde la línea de órdenes

1. Obtener una copia de HTNP2CL

Para obtener una copia del planificador HTNP2CL, escriba a alguna de las direcciones de correo electrónico de los autores: o.garcia@iactive.es, l.castillo@decsai.ugr.es, faro@decsai.ugr.es, f.palao@iactive.es, o t.garzon@iactive.es.

2. Invocación del planificador

La sintaxis para invocar al planificador HTNP2CL desde la línea de órdenes es la siguiente[1]:

```
Sintaxis: ./htnp2cl [opciones] --domain_file (-d) <domain.pddl> --problem_file (-p) <problem.pddl>
```

Donde las opciones son las siguientes:

- --help (-h): Pantalla de ayuda.
- --debug (-g): Lanza el planificador en modo de depuración del dominio.
- --verbose (-v[<level>] i.e: -v1): Lanza el planificador en modo verbosidad, para imprimir por pantalla las distintas decisiones que va tomando el planificador. Hay tres niveles de verbosidad (1,2 o 3) que aumentan progresivamente el número de mensajes que se imprimen por pantalla. Por defecto es 2.
- --output_file (-o) {filename}: Escribe el plan resultante como un fichero de texto plano en el fichero pasado como argumento.
- --xml_file (-x) {filename}: Escribe el plan resultante en formato xml para su posterior postproceso en el fichero pasado como argumento.
- --expansions_limit <number> : Número máximo de expansiones (aplicaciones de métodos) permitido. Sirve como límite del backtracking permitido durante la búsqueda.
- --depth_limit <number> : Nivel máximo de profundidad en el árbol de expansión permitido durante la búsqueda de un plan.
- --time_limit <number> : Tiempo máximo permitido para la búsqueda de un plan.

3. Invocación del planificador en modo servidor

El planificador puede ser lanzado como un servicio de planificación que puede ser invocado a través de internet usando el protocolo de comunicación XML-RPC. Generalmente es el módulo Metaserver el que se encarga de lanzar el planificador en este modo.

Cuando el planificador se lanza en este modo se le deben pasar dos argumentos más.

1 Versión 0.705 21 de Mayo del 2007.

- --port (-p) <number>: Puerto en el cual el planificador quedará escuchando a la espera por defecto es el 8080.
- --key (-k) <string>: Es una clave o llave que el servicio que levanta el planificador debe establecer. El planificador no responderá a los requerimientos de servicio a través de su API a no ser que la clave sea correcta, de esta forma se garantiza que ningún servicio no autorizado (que no conozca la clave) pueda acceder al plan resultante.

La API XML-RPC de HTNP2CL es plenamente funcional pero aún está en fase experimental y probablemente cambie en las siguientes versiones. Existen métodos para lanzar el planificador en modo de depuración en modo remoto y controlar dicha depuración pero en esta sección y también para obtener todo tipo de información acerca del plan y estadísticas de planificación. Nos limitaremos a describir las funciones de la API fundamentales.

void **eofPlanProcess**::execute(XmlRpcValue& params, XmlRpcValue& result)

Comprueba si el planificador ha terminado su ejecución.
Lee el estado en el que se encuentra el planificador.
En el caso de que el planificador haya sido lanzado con una clave (key) para el acceso seguro, hay que pasarla como primer argumento.
Los estados en los que se puede encontrar el planificador son los siguientes:
1 = Se encontró un plan
0 = El planificador está buscando una solución
-1 = Se detectó un error
-2 = Error en la llave, llave incorrecta

void **getXMLPlan**::execute(XmlRpcValue& params, XmlRpcValue& result)

Devuelve información del plan obtenido en formato XML

void **finalize**::execute(XmlRpcValue& params, XmlRpcValue& result)

Manda una señal de finalización al planificador.

void **readOStream**::execute(XmlRpcValue& params, XmlRpcValue& result)

Lee el flujo de salida donde el planificador escribe los mensajes con información.
El primer argumento es la key (clave).
Devuelve una cadena con el contenido del flujo.