EXAMEN PARCIAL BLOQUE 2 PLANSYS2

PREGUNTA 1

¿Qué es un sistema de planificación?

a. Es uno o varios algoritmos de planificación que resuelven planes especificados en PDDL.

b. Es una aplicación que permite resolver misiones en robot en un escenario complejo específico usando planificación.

c. Es un framework para la creación y ejecución de planes en robótica.

d. Es un estándar que define cómo especificar problemas y dominios PDDL.

PREGUNTA 2 (PREGUNTA ANULADA)

¿En qué parte de PlanSys2 se cargan los plugins de los planificadores (POPF, Optics, …) para resolver planes?

a. Planner Expert.

b. Problem Expert.

c. Domain Expert.

d. Executor.

e. En ningún sitio de PlanSys2. No existen plugins de planificadores.

PREGUNTA 3

Una aplicación que usa PlanSys2 está compuesta por …

a. Un dominio PDDL y la implementación de sus acciones.

b. Un controller (obligatorio) y la implementación de las acciones.

c. Un controller, la terminal de PlanSys2, y un dominio PDDL y la implementación de sus acciones.

d. Un controller (opcional), un dominio PDDL y la implementación de sus acciones.

PREGUNTA 4

La terminal de PlanSys2 …

a. Es desde donde despegan la implementación de las acciones.

b. Es una aplicación que te permite interactuar con PlanSys2 para obtener información y hacer peticiones a cada uno de sus componentes, usando comandos de texto.

c. Es el API de cada uno de los módulos de PlanSys2, codificados en el interfaz del que heredan cada uno de sus clientes.

d. Es el estado final en el que queda PlanSys2 tras resolver y ejecutar un plan.

PREGUNTA 5

get problem predicates

a. No me dice nada.

b. Permite ver los predicados que hay definidos en el Domain Expert.

c. Permite ver los argumentos (nombre y tipo) de un predicado.

d. Permite ver los predicados que hay definidos en el Problem Expert.

PREGUNTA 6

¿Dónde se usan Behavior Trees en PlanSys2?

a. El Executor codifica los planes como Behavior Trees para ejecutarlos. Las acciones también pueden ser implementadas como Behavior Trees.

b. Las acciones también pueden ser implementadas como Behavior Trees.

c. El Executor codifica los planes como Behavior Trees para ejecutarlos. Las acciones también pueden ser implementadas como Behavior Trees. Nav2 usa internamente Behavior Trees para definir cómo navegar.

d. El Executor codifica los planes como Behavior Trees para ejecutarlos.

e. Realmente no se usan.

PREGUNTA 7

¿Cómo se implementan las acciones que contienen un plan, conforme a un dominio PDDL?

a. Los Action Performers son LifeCycle Nodes que se activan cuando le toca ejecutar la acción que implementan, recibiendo los argumentos que figuran en el plan.

b. Ninguna es correcta.

c. Un Controller contiene la implementación de las acciones de una aplicación.

d. Los Action performers son programas que se ejecutan cuando le toca ejecutar la acción que implementan.

PREGUNTA 8

¿Qué hace un Controller?

a. Inicializa el conocimiento, implementa las acciones del dominio, establece los goals, pide planes y controla la ejecución de estos planes.

b. Inicializa el conocimiento, establece los goals, pide planes y controla la ejecución de estos planes, replanificando si es necesario.

c. Inicializa el conocimiento, establece los goals, pide planes y controla la ejecución de estos planes.

d. Implementa las skills de actuación que usan las acciones.

PREGUNTA 9

¿Quién ejecuta una acción?

a. El primer Action Performer que responde al Executor cuando solicita la ejecución de una acción en el /actions_hub.

b. Ninguna es correcta.

c. El Action Performer que especifica el Executor el /actions_hub.

d. El mejor Action Performer que responde al Executor cuando solicita la ejecución de una acción en el /actions_hub.

PREGUNTA 10
¿Qué es esto?

```
class MoveAction : public plansys2::ActionExecutorClient
{
public:
	MoveAction()
	: plansys2::ActionExecutorClient("move", 250ms)
	{
		progress_ = 0.0;
	}
private:
	void do_work()
	{
		if (progress_ < 1.0) {
			progress_ += 0.02;
			send_feedback(progress_, "Move running");
		} else {
			finish(true, 1.0, "Move completed");
			progress_ = 0.0;
			std::cout << std::endl;
		}
		std::cout << "\r\e[K" << std::flush;
		std::cout << "Moving ... [" << std::min(100.0, progress_ * 100.0) << "%] " << std::flush;
	}
	float progress_;
};
```

a. Parte de un Controller.
b. La implementación de un Action Performer.
c. Esto no tiene relación con PlanSys2.
d. Parte de un plugin de un planificador.

PREGUNTA 11
¿Qué es “arg3” en esto?

```
<root BTCPP_format="4" main_tree_to_execute = "MainTree">
	<BehaviorTree ID="MainTree">
		<Sequence name="root_sequence">
			<OpenGripper name="open_gripper"/>
			<ApproachObject name="approach_object"/>
			<CloseGripper name="close_gripper"/>
			<Move name="move" goal="{arg3}"/>
			<OpenGripper name="open_gripper"/>
		</Sequence>
	</BehaviorTree>
</root>
```

a. El tercer argumento del nodo Move.
b. No significa nada. Es incorrecto.
c. El tercer argumento de la acción que se implementa con este Behavior Tree.
d. El tercer argumento con el que se llama a este controlador.

SOLUCIONES

PREGUNTA 1. OPCIÓN CORRECTA: c
a. FALSO, también poseen una infraestructura para gestionar dominios, problemas, etc.
b. FALSO, no está restringido a aplicación / escenarios específicos ya que es una herramienta general.
d. FALSO, describe el funcionamiento de PDDL pero no de un sistema de planificación.

PREGUNTA 2. OPCIÓN CORRECTA: a
b. FALSO, el Problem Expert gestiona los predicados y objetos del problema actual pero no los planificadores.
c. FALSO, el Domain Expert maneja la definición del dominio PDDL pero no los plugins.
d. FALSO, el Executor se encarga de ejecutar el plan pero no de generarlo.
e. FALSO, los plugins de planificadores sí existen y son fundamentales en PlanSys2.

PREGUNTA 3. OPCIÓN CORRECTA: d
a. FALSO, no menciona el controller aunque sea opcional.
b. FALSO, el controller no es obligatorio.
c. FALSO, la terminal de PlanSys2 es una herramienta externa que no forma parte de la aplicación.

PREGUNTA 4. OPCIÓN CORRECTA: b
a. FALSO, las acciones se implementan en el código pero no despegan desde la terminal.
c. FALSO, la terminal no es una API sino una interfaz de usuario.
d. FALSO, no tiene ninguna relación con el estado final del sistema.

PREGUNTA 5. OPCIÓN CORRECTA: d
a. FALSO, es un comando válido en la terminal de PlanSys2.
b. FALSO, los predicados que hay definidos en el Domain Expert se pueden ver con get domain predicates.
c. FALSO, los argumentos de un predicado se pueden ver con get domain predicate details <name>.

PREGUNTA 6. OPCIÓN CORRECTA: a
b. FALSO, no menciona el rol clave del Executor (transformar el plan en un Behavior Tree).
c. FALSO, Nav2 no es parte de PlanSys2 ya que es una herramienta externa.
d. FALSO, no menciona que las acciones también pueden ser Behavior Trees.
e. FALSO, los Behavior Trees son fundamentales en PlanSys2 para la ejecución de planes.

PREGUNTA 7. OPCIÓN CORRECTA: a
b. FALSO, la opción a es correcta.
c. FALSO, el Controller gestiona objetivos y ejecución pero no implementa acciones (Action Performers).
d. FALSO, no menciona que los Action Performers son LifeCycle Nodes.

PREGUNTA 8. OPCIÓN CORRECTA: b
a. FALSO, el Controller no implementa acciones.
c. FALSO, no menciona la replanificación la cual es una función clave del Controller.
d. FALSO, las skills de actuación no son parte del Controller en PlanSys2.

PREGUNTA 9. OPCIÓN CORRECTA: a
b. FALSO, la opción a es correcta.
c. FALSO, no especifica que hay que usar el primer Action Performer que responde al Executor.
d. FALSO, hay que usar el primer Action Performer que responde al Executor pero no el mejor.

PREGUNTA 10. OPCIÓN CORRECTA: b
a. FALSO, el Controller no implementa acciones.
c. FALSO, la clase ActionExecutorClient es específica de PlanSys2.
d. FALSO, los plugins de los planificadores no usan la clase ActionExecutorClient.

PREGUNTA 11. OPCIÓN CORRECTA: c

a. FALSO, confunde la sintaxis del Behavior Tree con la estructura de la acción PDDL.

b. FALSO, ignora el vínculo entre Behavior Tree y PDDL en PlanSys2.

d. FALSO, el controlador es irrelevante en la ejecución de acciones.