

UAEM Centro Universitario UAEM Zumpango.

Ingeniería en Computación.

AGENTES

INTELIGENTES

Inteligencia Artificial

Profesor: Asdrúbal López Chau

Alumna: Keren Mitsie Ramirez Vergara.

01 Octubre 2021

# GLOSARIO

## A

**Agente:** Sistema informático que está situado en algún entorno y que es capaz de actuar de forma autónoma en este entorno para cumplir sus objetivos de diseño

**Agentes óptimos:** Son agentes que utilizan funciones que se pueden implementar en una computadora real

**Agentes puramente reactivos:** Son agentes que deciden que hacer sin hacer referencia a su historial, ellos basan su toma de decisiones enteramente en el presente, sin ninguna referencia al pasado

**Ambiente de tareas:** un entorno de tareas está diseñado para ser un par  $(Env, W)$  donde  $Env$  es un entorno y  $W: R \rightarrow \{0, 1\}$ . También se le conoce como predicado sobre ejecuciones.

## C

**Capacidad efectórica:** capacidad de un agente para modificar sus entornos

## E

**Entorno episódico:** entorno donde el agente actúa dependiendo de una serie de episodios discretos, además, es un vínculo entre el desempeño del agente en diferentes episodios. Un ejemplo es el sistema de clasificación de correo.

**Equidad:** es cuando un agente toma una decisión que parece completamente razonable en un contexto local pero tiene efectos indeseables cuando se consideran en el contexto de toda la historia del sistema.

## H

**Habilidad social:** capacidad de un agente de interactuar con otros agentes (y posiblemente humanos) para satisfacer sus objetivos de diseño.

# I

**Interacción en tiempo real:** es una interacción en el cual el tiempo juega un papel importante en la evaluación del desempeño de un agente. Existen diferentes tipos de interacción en tiempo real:

- a) Un agente debe tomar una decisión dentro de un límite de tiempo especificado
- b) Un agente debe tomar un estado de las cosas tan rápidamente como sea posible
- c) Un agente debe repetir la tarea con la mayor frecuencia posible

# M

**Multiagente:** es aquel agente que tiene más agentes para que pueda observar el ambiente completo. Un multiagente debe tener por lo menos un hilo de control.

# O

**Objeto activo:** aquel objeto que abarca su propio hilo de control. Los objetos activos son generalmente autónomos, lo que significa que pueden exhibir algún comportamiento sin ser operados por otro objeto

**Obstáculos en agentes:** grupo de celdas de cuadrícula inamovibles

# P

**Postcondición:** representa lo que será cierto del entorno del programa después de que el programa termine.

**Postura de diseño:** es cuando utilizamos el conocimiento de qué propósito se supone que debe cumplir un sistema para predecir cómo se comporta

**Postura física:** es cuando se comienza con la configuración original de un sistema, y luego usar las leyes de la física para predecir cómo se comportará este sistema.

**Postura intencional:** la postura intencional justifica cuando la complejidad de un sistema impide que pueda predecir su comportamiento

**Precondición:** representa lo que debe ser cierto en el entorno del programa

**Proactividad:** capacidad de un agente de exhibir un comportamiento dirigido a objetivos al tomar la iniciativa para satisfacer sus objetivos de diseño.



# R

**Reactividad:** capacidad de un agente de percibir su entorno y responder de manera oportuna a los cambios que se produzcan en el mismo con el fin de satisfacer su diseño objetivos.

# S

**Sistema experto:** sistema que es capaz de resolver problemas o dar asesoramiento en algún ámbito rico en conocimientos

**Sistema funcional:** sistema que toma alguna entrada y eventualmente produce alguna salida. Uno de los atributos clave de estos sistemas funcionales es que terminan. Las propiedades pueden entenderse en términos de precondiciones y postcondiciones.

**Sistemas reactivos:** son sistemas que no pueden describirse adecuadamente por visión relacional o funcional.

**Sistemas tradicionales:** sistemas cuyo comportamiento puede predecirse mediante el método de atributos, creencias, deseos y perspicacia racional

# T

**Tareas de mantenimiento:** tareas que mantienen el estado de cosas

**Tareas predicadas:** cuando se habla de tareas, se usa en términos de utilidades, son aquellas en la que actúa la función de utilidad como predicado se Sobrepasa. Formalmente, diremos una función de utilidad

**Timeworld:** sistema que se propuso principalmente como un entorno experimental, entorno para evaluar arquitecturas de agentes. El timeworld nos permite examinar varias capacidades de los agentes. El desempeño de un agente se define como:

$$U(r) = \frac{\text{número de agujeros llenados en } r}{\text{número de agujeros que aparecieron en } r}$$

**Tareas de logros:** es una tarea que se especifica mediante una serie de estados de meta, el agente se requiere para lograr uno de estos estados objetivos

**Transformador de estado:** función que mapea una cámara, con un conjunto de posibles riesgos ambientales, los que podrían resultar de realizar la acción

# U

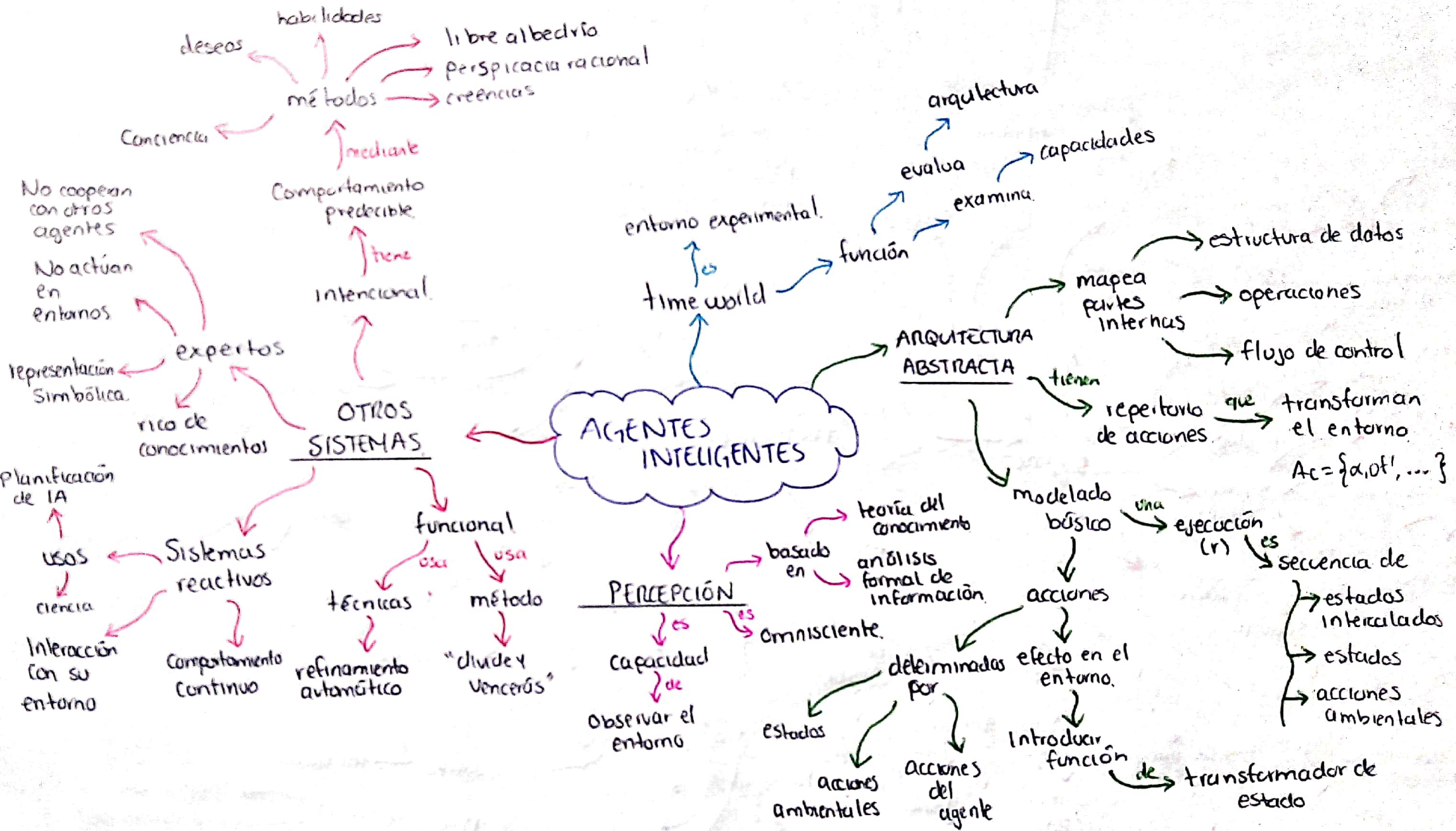
**Utilidad:** es un valor numérico que representa cuán "bueno" es el estado. Cuanto mayor sea la utilidad, mejor.

✓

Visión relacional: Son programas considerados como funciones que van de un estado inicial a un estado terminal.







la tecnología de agentes fue conocida como especialización de programación orientada a objetos

1995

Modelo: basado en reglas

Las reglas son abstracciones de primer orden que pueden describir un comportamiento

CHROMAR

Modelo: BDI

Agentes que reaccionan a los eventos del sistema ejecutando acciones en el entorno.

JASON

Modelo: basado en reglas

Lenguaje de programación de agente declarativo que utiliza una base de conocimientos.

GOAL

Modelo: BDI y OPP

Programación de agentes inteligentes en XML y JAVA.

Características:  
→ Infraestructura de tiempo de ejecución  
→ Simulaciones.

JADEX.

→ redes superpuestas

Modelo: BDI, HTN

Agrega capacidad de planificación de IA a los agentes

Tiene un planificador capaz de manejar información temporal

PLACE

Modelo: DSL

Se centra en comparar plataformas paralelas que admiten múltiples agentes simulaciones y

JANDEL

selección en recursos.

1990

1993

Se introdujo por primera vez la programación orientada a agentes.

JADE

Modelo: FIPA

- Desarrollo de agentes peer to peer.
- Los sistemas pueden distribuirse entre máquinas con diferentes sistemas operativos
- La abstracción proporciona ejecución y composición de tareas.

JACAMO

- Se compone de tres tecnologías
  - a) Jason: programar en nivel de agente
  - b) ChrtAg: responsable del nivel del medio ambiente
  - c) Moise → nivel de la organización

Modelo: BDI

GWENDOLEN

Modelo: BDI

- Creado para respaldar la verificación del agente desde cero
- Es limitado en las funciones que puede admitir.

SARL

Modelo: DSL

- Proporciona un lenguaje extensible que está equipado con la cantidad mínima de conceptos.
- Tiene como objetivo proporcionar abstracciones para la concurrencia, distribución, interacción, autonomía.

JADESCRIPT

Modelo: DSL Scripting.

- Programación escalable.
- Características
  - a) expresiones lambda
  - b) definición de múltiples planos y reglas
  - c) ejecución en paralelo.
  - d) Variables seguras para subprocesos.

PLASA

Modelo: Wirt-look-computer-move.

- Rápida implementación de aplicaciones cooperativas
- Lenguaje de programación de alto nivel

RIMAS

Modelo: database-centric, CPS.

- Enfoque cerrado en la base de datos
- Acoplamiento del entorno Matlab y el lenguaje de la BD SQLite.

2000 - 2020

(desde creación, antes de 2015 hasta recientes actualizaciones)

1990



c) Investigar y explicar brevemente en qué consisten los modelos de la tabla 1.

## Modelo BDI

De las siglas Belief, Desire, Intention está caracterizada por el hecho de que los agentes que la implementan están dotados de los estados mentales de creencias, deseos e intenciones. Los agentes y en particular los agentes BDI incorporan componentes que permiten el desarrollo de sistemas que se integran adecuadamente en el mundo real.

- Las creencias representan el conocimiento que se tiene del entorno
- Un objetivo representa un estado final deseado a los que se quiere lograr
- La intención es una secuencia de pasos hacia la construcción de un deseo

Las relaciones entre las creencias, deseos e intenciones pueden ser:

- a) Compatibilidad entre creencias y objetivos.
- b) Compatibilidad entre objetivos e intenciones.
- c) Las intenciones conducen a acciones
- d) Relación entre creencias e intenciones
- e) Relación entre creencias y objetivos
- f) No hay retrasos infinitos.

## Modelo FIPA (Foundation for Intelligent Physical Agents)

Define un modelo para la administración de agentes para que pueda existir, operar y ser gestionados, estableciendo así una referencia lógica para la creación, registro, localización, comunicación, migración y operaciones entre agentes.

Sus componentes son:

- plataforma de agentes: infraestructura en la que se establecen los agentes, pueden encontrarse distribuida en varios ordenadores, los agentes que residen en la misma no tienen por qué encontrarse en el mismo equipo.
- Agente: proceso computacional que reside en la plataforma de agentes puede encontrarse distribuida en varios ordenadores
- Facilitador de directorio: mantiene una lista completa y precisa de los agentes existentes, proporciona información actualizada sobre los servicios que estos ofrecen.
- Sistema de gestión de agentes (AMS): ofrece servicios como creación, eliminación y control de los cambios de estado entre agentes.
- Sistema de transporte de mensajes (MTS): transporta mensajes FIPA-ACL entre agentes.

## Modelo basado en reglas:

Permite la creación de sistemas con base de conocimiento de la que se contiene variables y un conjunto de reglas que definen el problema. El monitor de inferencia es capaz de extraer conclusiones aplicando métodos de la lógica clásica sobre esta base. Una regla es una proposición lógica que relaciona dos o más objetos del dominio. e incluye dos partes, la premisa y la conclusión.



## Modelo DSL:

Domain Specific languages, modelo especializado en modelar o resolver un conjunto específico de problemas. Usa lenguajes generalistas o diversos lenguajes de propósito específico, para resolver los problemas que surgen en un agente.

Los DSL intercambian generalidad por expresividad en un dominio limitado. Ofrece notaciones y construcciones orientadas a un dominio particular, presentando en la eficiencia del agente.

Los DSL se pueden clasificar en:

- a) Internos: un determinado lenguaje anfitrión es usado para darle la apariencia de otro lenguaje concreto.
- b) Externos: Tienen su propia sintaxis y es necesario un parser para poder ejecutarlos.

## Modelo wait - Look - compute - Move.

Modelo para un conjunto de robots autónomos, donde cada robot es ubicado en un gráfico o entorno. Cada robot mira repetidamente su entorno y obtiene una instantánea información (vértice) y luego se mueve hacia él. Los robots tienen luces visibles, lo que permite comunicar más información de su posición actual, se mueven de forma asincrónica, es decir, se mueven de forma que tienen una velocidad arbitraria.

## Modelo de interacción.

Se caracteriza en términos de un álgebra de comportamientos que es un álgebra continua con aproximación y dos operaciones: elección no determinista y prefijo. Las funciones continuas arbitrarias se pueden utilizar como funciones de inserción y utilizamos funciones definidas que sean calculables. La transformación de los comportamientos ambientales también definen el nuevo tipo de equivalencia del agente.

## Modelo Database-centric

Es un modelo basado en las bases de datos, alrededor de la base de datos se desarrolla una capa de persistencia de la que depende el nivel de conocimientos para lograr un objetivo, esta capa transforma el proceso de datos en toma de decisiones.

## Modelo CPS:

Modelos basados en sistemas ciberfísicos, combinan los dispositivos digitales y analógicos, interfaces, redes, sistemas informáticos y similares con el mundo físico natural. El modelado y simulación le proporciona conocer el sistema ciberfísico.

## Modelo ODP:

Modelo que trabaja con objetos del mundo real, cada componente es una unidad autónoma, que se relacionan entre sí. Cada componente tiene dos elementos: la característica y el comportamiento. Se utilizan términos como Clase, métodos, herencia, polimorfismo, etc.



d) Menciona 4 aplicaciones recientes (últimos 5 años) de los agentes mencionados en el artículo.

## NetLogo

Primera implementación del entorno de modelado interactivo para desarrollar modelos acoplados de aguas subterráneas (Flowlogo), ofrece una forma sencilla de representar la evolución del comportamiento del agente con las condiciones del agua subterránea.

## MICROGRID

Utiliza tecnologías para el control de Microgrid i atiende en el control y funcionamiento de las microredes para problemas de optimización energética. Adopta la reducción de emisión de CO<sub>2</sub> para el medio ambiente.

## Agente basado en reglas para COVID-19.

El objetivo de este agente es desarrollar y analizar un modelo de enfermedad que represente la propagación de la enfermedad COVID-19. El agente debe de modelar

- 1) viabilidad de la tasa de interacción entre agentes
- 2) la estructura de la red de contactos en personas

## Agentes como microservicios

Agentes que utilizan el desarrollo de aplicaciones de IoT utilizando agentes y líneas de productos de software en sistemas de gestión.

Representa el soporte novedoso para la administración de aplicaciones en esta área.