# Sistemas Multi-Agentes

## Metodología: Prometheus y PDT

Dr. Alejandro Guerra-Hernández

**Instituto de Investigaciones en Inteligencia Artificial**
Universidad Veracruzana

*Campus Sur, Calle Paseo Lote II, Sección Segunda No 112,*
*Nuevo Xalapa, Xalapa, Ver., México 91097*

mailto:aguerra@uv.mx
https://www.uv.mx/personal/aguerra/sma

Maestría en Inteligencia Artificial 2024

Universidad Veracruzana

# La metodología Prometheus

- Prometheus [1] es una metodología de ingeniería de software basada en agentes (AOSE) [2].
- Aunque se usa principalmente en la academia, está asociado a la plataforma de desarrollo JACK [1].

Universidad Veracruzana

[1] https://aosgrp.com.au/jack/

# Características

- Es detallada, provee una guía de cómo ejecutar los diferentes pasos que conforman el proceso Prometheus.
- Soporta el diseño de agentes basados en metas y planes.
- Cubre actividades desde la especificación de requerimientos hasta el diseño detallado del sistema.
- Tiene un software de soporte asociado, PDT.
- Está dirigido a desarrolladores industriales y estudiantes de licenciatura, no investigadores y estudiantes de posgrado.

Universidad Veracruzana

# ¿En qué conciste la metodología Promethus?

- Una descripción de conceptos para el diseño de agentes.
- Un proceso.
- Un conjunto de notaciones para capturar los diseños.
- Diversas técnicas que aconsejan como llevar a cabo los pasos del proceso Prometheus.

Universidad Veracruzana

# Conceptos

- Un agente inteligente es un *software* situado, autónomo, reactivo, pro-activo, flexible, robusto y social [3].
- En tanto que situado, la interfaz de un agente se expresa en términos de percepciones que proveen información del ambiente y y acciones que lo modifican.
- En tanto que pro-activo, un componente central de un agente son sus metas.
- En tanto que reactivo, un agente responde a la ocurrencia de eventos significativos, e.g., percepciones, mensajes, eventos internos como agregar una meta o una creencia, etc.
- En tanto que sociales, los agente están sujetos a compromisos, normas y equipos.

Universidad Veracruzana

# Otros conceptos

- Las creencias son el estado almacenado del agente. Son importantes al menos que el agente sea puramente reactivo.
- Los planes son importantes porque no hay tiempo para planear a partir de primeros principios.

Universidad Veracruzana

# Fases de Diseño

- La metodología consiste de tres fases de diseño:
  - Especificación del sistema basada en metas y escenarios. La interfaz del sistema con su ambiente se describe en términos de acciones, percepciones y datos externos. También se definen las funcionalidades del sistema.
  - Diseño de arquitectura basada en los tipos de agentes en el sistema. Los escenarios de desarrollan como protocolos de interacción para producir un diagrama de la vista general del sistema.
  - Diseño detallado de los agentes en términos de capacidades, datos, eventos y planes. Los diagramas de proceso se usan para conectar los protocolos de interacción y los planes.

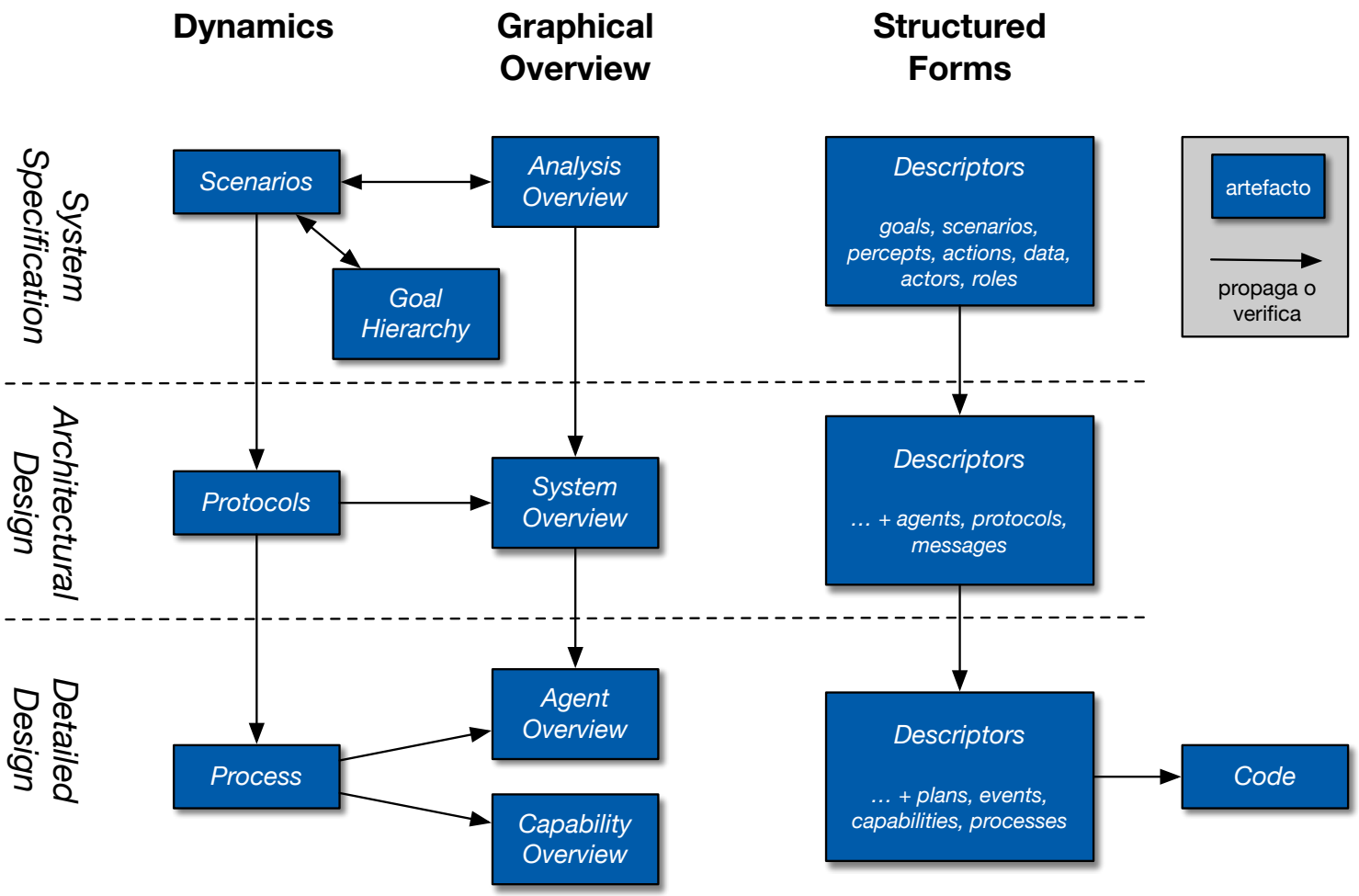
Universidad Veracruzana

# Artefactos

- Cada fase incluye artefactos de diseño que abarcan:
  - Dinámica del sistema
  - Modelos gráficos de diseño estructural
  - Descriptores detallados de las entidades del sistema.

Universidad Veracruzana

# Gráficamente




Universidad Veracruzana

# Información estructurada

- Muchos de estos artefactos son altamente estructurados.
- Esto permite:
  - La propagación automática de información entre fases y entre diversos aspectos de cada fase.
  - la verificación de consistencia que garantiza la coherencia interna de diseño.

Universidad Veracruzana

# Especificación del sistema

- Inicia con un esbozo del sistema que puede ser una breve descripción del mismo.
- Luego, se definen los requerimientos del sistema en términos de:
  - Las metas del sistema
  - Los escenarios de casos de uso
  - Las funcionalidades
  - La interfaz del sistema con su ambiente, en términos de acciones y percepciones.
- Estos requerimientos no de procesan secuencialmente, se trata de un proceso iterativo.

Universidad Veracruzana

# Metas

- Partimos de un conjunto de metas iniciales del sistema, i.e., las planteadas en la descripción de alto nivel del sistema.
- Las metas iniciales de desarrollan preguntándonos ¿Cómo puede satisfacerse cada una de estas mentas? Lo que da lugar a sub-metas.
- El conjunto de metas se revisa para identificar sub-metas comunes.
- Las metas se representa usando un diagrama de metas que muestra la jerarquía resultante.

Universidad Veracruzana

# Funcionalidades

- Las funcionalidades son agrupamientos de metas que proveen trozos de conducta coherente.
- Incluyen un grupo de metas relacionadas, percepciones relevantes, acciones a ejecutar y los datos que se usarán.
- Suelen terminar especificadas como capacidades de los agentes.
- Suelen revisarse considerando los tipos de agente en el sistema (diseño de arquitectura).
- La información extra se captura mediante descriptores, e.g., eventos disparadores.

Universidad Veracruzana

# Escenarios

- Son descripciones de una secuencia de eventos asociados al logro de una meta o a la respuesta a un evento en particular.
- Se describen usando un nombre, una descripción y un evento disparador.
- Su parte central es una secuencia de pasos.
- Cada paso está asociado a una funcionalidad, el nombre del paso y su tipo (acción, percepción, meta, escenario u otra).
- Pequeñas variaciones (alternativas) también pueden definirse en un escenario.

Universidad Veracruzana

# Ambiente

- Se define en términos de percepciones, acciones y datos externos.
- Las percepciones consideran su procesamiento para extraer información útil.
- La especificación del ambiente es más complicada para agentes situados en ambientes físicos, i.e., robots.

Universidad Veracruzana

# Diagrama de vista general del análisis

- Permite la identificación de actores y sus interacciones con el sistema, en forma de percepciones, lecturas de fuentes de información y acciones;

Universidad Veracruzana

# Diseño de la arquitectura

- Determinar los tipos de agentes agrupando funcionalidades bajo consideraciones de acoplamiento y conocimiento entre estos;
- Desarrollar los protocolos de interacción derivados de los escenarios de casos de uso;
- Desarrollar el diagrama general del sistema que captura los tipos de agentes, las fronteras del sistema y sus interfaces.

Universidad Veracruzana

# Tipos de agentes

- Cada tipo de agente debe ser cohesivo y el acoplamiento entre agentes debe ser bajo.
- Un tipo de agente se forma al combinar una o más funcionalidades.
- Algunas razones a considerar cuando se agrupan agentes incluyen:
  - Si dos funcionalidades están claramente relacionadas, tiene sentido agruparlas en el mismo tipo de agente; Si no están claramente relacionadas, quizás no deberían agruparse en el mismo tipo.
  - Si dos funcionalidades requieren el mismo dato, quizás deberían agruparse.
- El diagrama de acoplamiento de datos es útil para esto.

Universidad Veracruzana

# Selección de diseños

- El proceso anterior resulta en un conjunto de diseños posibles.
- Una técnica para comparar estos diseños es usar el diagrama de conocidos para estimar el acoplamiento entre agentes.
- Una alta densidad de ligas en este diagrama implica un mayor grado de acoplamiento.

Universidad Veracruzana

# Descriptores de agentes

- Además de capturar:
  - La interfaz del agente
  - Las metas que puede resolver
  - Sus funcionalidades combinadas
  - Los protocolos que puede usar
- Fomentan pensar en el ciclo de vida de los agentes: Cuando serán creados y destruidos; y qué es necesario hacer en esos casos.

Universidad Veracruzana

# Interacciones entre agentes

- Usan diagramas de interacción y protocolos de interacción.
- Las notaciones usadas son simplificaciones de los diagramas de secuencia de UML.
- Los protocolos de interacción se definen usando AUML.
- Los diagramas de interacción se derivan de los escenarios de casos de uso.
- Estos diagramas capturan ejemplos de interacciones, en lugar de todas las posibles interacciones.

Universidad Veracruzana

# Protocolos de interacción

- Suelen combinar diferentes diagramas de interacción, i.e., variaciones del los escenarios.
- Otra técnica para su desarrollo consiste en preguntar ¿Qué más podría pasar aquí? en cada punto de una secuencia de interacción.
- Se combinan con descriptores para protocolos y mensajes.
- El diagrama de vista general del sistema captura todo esto.

Universidad Veracruzana

# Diseño detallado

- Definir los detalles internos de cada agente en términos de capacidades (y en algunos casos, directamente en términos de eventos, planes y datos).
- Se pueden usar los diagramas generales de agente y capacidades. Un diagrama general de capacidad captura la estructura de los planes para la capacidad y los eventos asociados con estos planes.
- El comportamiento dinámico se describe mediante los diagramas de procesos basados en los protocolos de interacción.
- Los detalles de las capacidades se desarrollan en términos de otras capacidades, eventos, planes y datos. Se debe garantizar un conjunto de planes que cubran adecuadamente las metas.

Universidad Veracruzana

# Capacidades

- En este contexto, las capacidades son un mecanismo de estructuración similar a los módulos.
- Pueden contener planes, datos y eventos.
- También pueden contener otras capacidades, permitiendo así estructuras jerárquicas.
- Se puede partir de las capacidades para cada funcionalidad agrupada en el tipo de agente. Se puede:
  - Mezclar capacidades similares y pequeñas.
  - Separar capacidades que son demasiado grandes.
  - Agregar capacidades que se corresponden con librerías.

Universidad Veracruzana

# Diagrama General de Agente

- Muestra la estructura de cada agente.
- Es similar al diagrama general del sistema, pero no contiene agentes; ni suele contener protocolos.
- En cambio, contiene capacidades y, algunas veces, planes.

Universidad Veracruzana

# Diagramas de procesos

- Los protocolos de interacción son globales, en el sentido de que muestran las interacciones a vuelo de pájaro.
- Aquí se desarrollan los diagramas de proceso basados en los protocolos de la fase de diseño arquitectural.
- Cada protocolo de interacción tiene múltiples diagramas de procesos, correspondientes al punto de vista de diversos agentes.
- Se usan extensiones a los diagramas de actividad UML.
- Se asume que los agentes serán implementados en una plataforma que soporte planes, eventos disparadores y metas, p. ej., Jason.

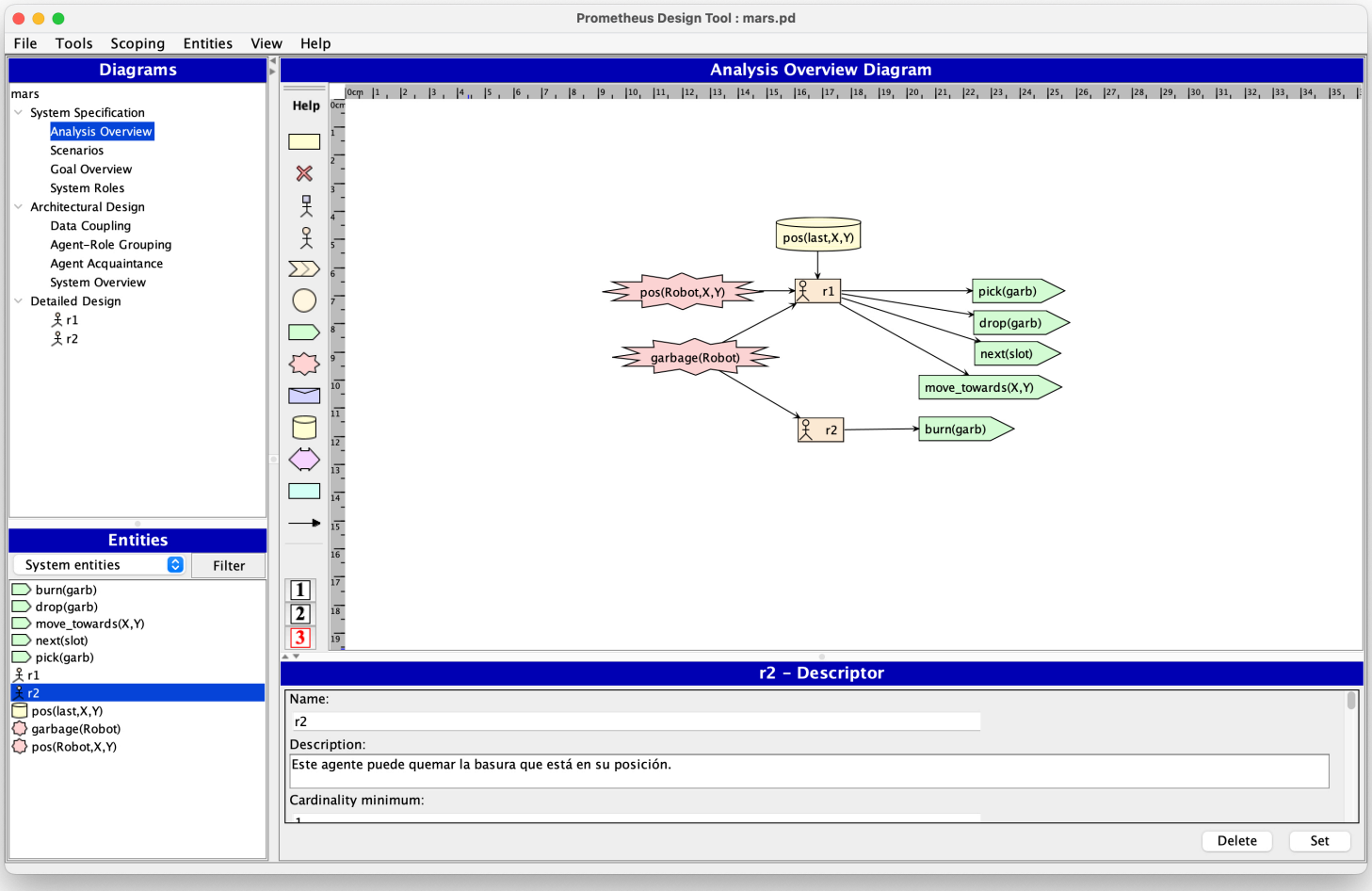
Universidad Veracruzana

# Herramienta de soporte

- Dada la naturaleza iterativa del proceso Prometheus, tanto ente fases como dentro de las mismas, el diseño suele cambiar constantemente.
- Cuando el diseño es muy grande, resulta difícil garantizar que las consecuencias de un cambio se propagan adecuadamente.
- La herramienta PDT da soporte a la metodología Prometheus, proporcionando una interfaz gráfica para:
  - Verificación entre los diagramas;
  - Propagación automática de elementos de diseño, cuando es posible y apropiado; y
  - Asistencia en la búsqueda de nombres.

Universidad Veracruzana

# Ventana principal



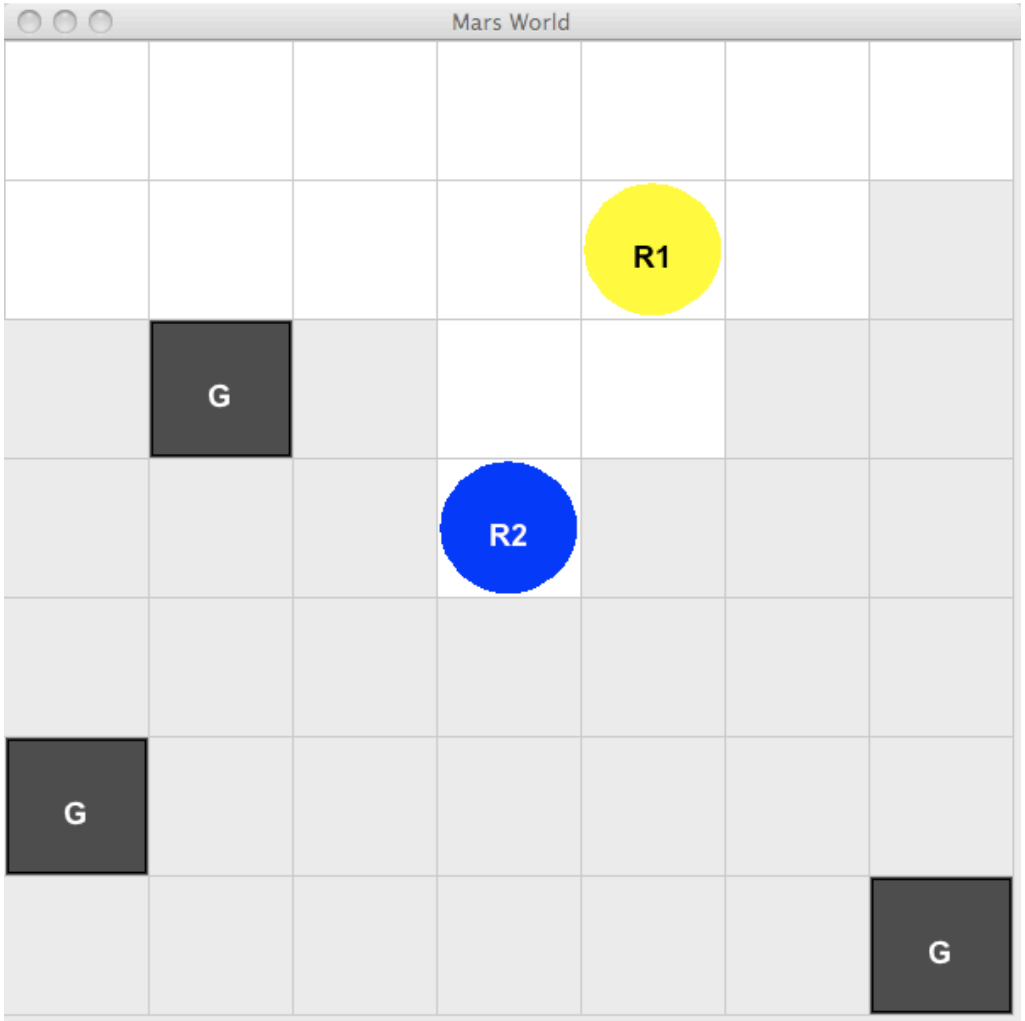

Universidad Veracruzana

# Verificación de consistencia

- Entidades existentes. Si una referencia a una entidad es creada entonces la entidad también es generada si no existe, y cuando una entidad es borrada todas sus referencias son eliminadas.
- Nombres. No es posible asignar el mismo nombre a dos entidades.
- Errores simples. Solamente son permitidas conexiones válidas entre entidades.
- Inconsistencia entre los niveles de detalle. por ejemplo, si un agente se especifica para que solo lea un conjunto de creencias, entonces no puede contener un plan que escriba en ese conjunto de creencias.

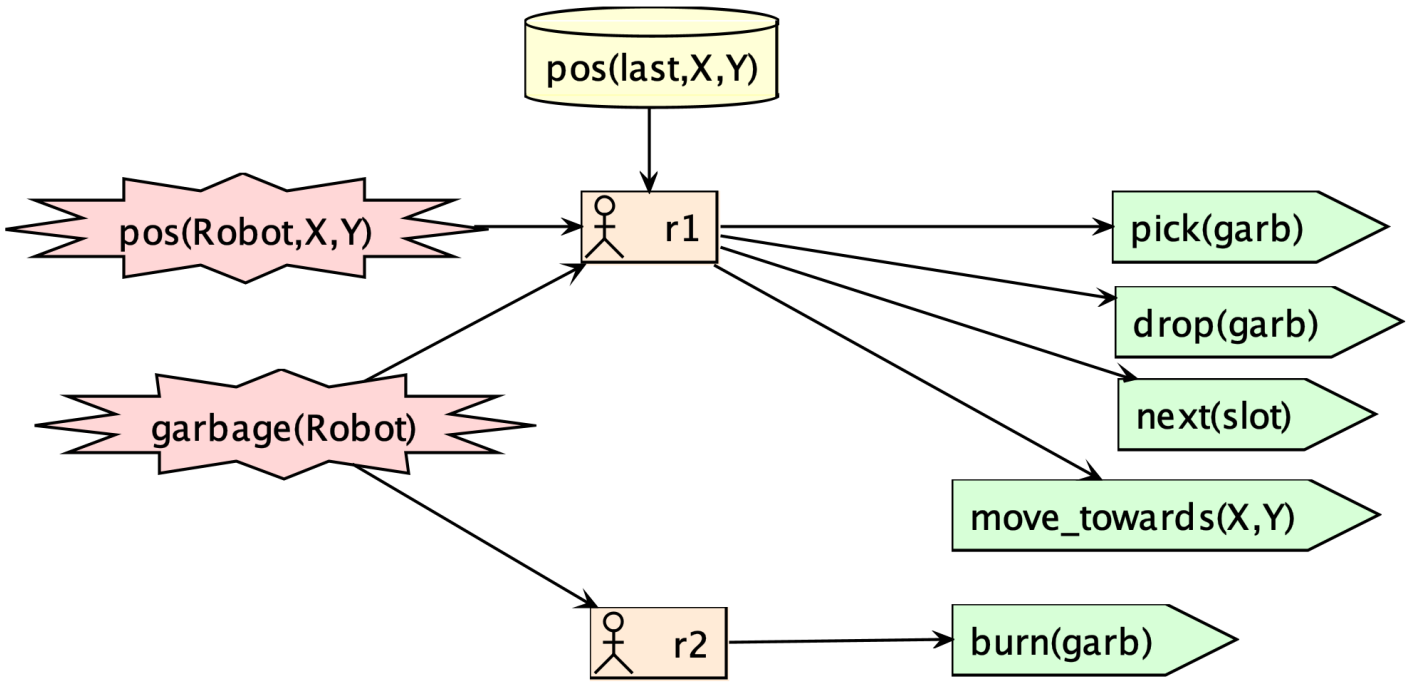
Universidad Veracruzana

# Limpiadores en marte




Universidad Veracruzana

# Diagrama general




Universidad Veracruzana

# Bibliografía


[1] L Padgham y M Winikoff. *Developing intelligent agent systems: A practical guide*. Wiley Series in Agent Technology. John Wiley & Sons, Ltd, 2004.

[2] O Shehory y A Sturm. *Agent-Oriented Software Engineering: Reflections on Architectures, Methodologies, Languages, and Frameworks*. Berlin, Germany: Springer-Verlag, 2014.

[3] M Wooldridge. *An Introduction to MultiAgent Systems*. 2nd. West Sussex, England: John Wiley & Sons, LTD, 2009.


Universidad Veracruzana