



**Eduardo Romero**

(eromero[at]conae.gov.ar)

**Comisión Nacional de  
Actividades Espaciales**

**AI Planning &  
Scheduling:  
Enfoques,  
Formalismos  
y Aplicaciones**

***Simposio de Doctorandos de Computación  
FaMAF - 6 Dic, 2010***

## **Presentación**

## **Planificación**

Introducción

Representación y algoritmos

## **Scheduling**

Descripción de los problemas

Algunas técnicas utilizadas

## **Operaciones de Misiones Satelitales**

## **Presentación**

## Planificación

Introducción

Representación y algoritmos

## Scheduling

Descripción de los problemas

Algunas técnicas utilizadas

## Operaciones de Misiones Satelitales

- Eduardo Romero (Yo):
  - Lic. Ciencias Matemáticas (Universidad de Buenos Aires)
  - Docencia en UBA/Inst.Gulich
  - Ingeniería del Segmento Terreno y Aplicaciones en CONAE
- Marcelo Oglietti (mi director):
  - Dr. Cs. Computación (Universidad La Sapienza - Roma)
  - Docencia en UBA/Inst.Gulich
  - Ingeniería del Segmento Terreno y Aplicaciones en CONAE

## **Presentación**

## **Planificación**

**Introducción**

Representación y algoritmos

## **Scheduling**

Descripción de los problemas

Algunas técnicas utilizadas

## **Operaciones de Misiones Satelitales**

- Planificar es el proceso abstracto de elegir y organizar acciones para cumplir un objetivo, basándose en los resultados esperados. Es necesario para actuar persiguiendo un fin determinado.
- *AI Planning & Scheduling*: rama de la inteligencia artificial que estudia la automatización de ese proceso en todas sus formas.

Algunos ámbitos donde aparece.

- Logística de transporte de objetos (materias primas, productos, etc).
- Líneas de producción.
- Desarrollo de proyectos.
- Autonomía de acción de robots.
- Y muchos más...

Qué tienen en común todos estos problemas?

El *dominio* sobre el que se planifica tiene:

- Posibles *estados* o evoluciones del sistema.
- *Acciones* (operadores) que pueden modificar esos estados.
- *Estado inicial* (situación de partida).
- *Objetivos* o condiciones que se desean ver satisfechas si el *plan* (secuencia de acciones) se ejecuta.



# Planificación - Introducción

*Representación básica como sistema de transición de estados*

*[Ghallab, Nau, & Traverso2004]*

Un *dominio de planificación* es una tupla,  $\Sigma = (\mathcal{S}, \mathcal{A}, \gamma)$ , donde

- $\mathcal{S}$  es el conjunto de estados.
- $\mathcal{A}$  es el conjunto de acciones.
- $\gamma : \mathcal{S} \times \mathcal{A} \rightarrow \mathcal{S}$  es la función de transición de estados.

# Planificación - Introducción

## Representación básica como sistema de transición de estados

Un *problema* de planificación es una tupla,  $\mathcal{P} = (\Sigma, s_0, \mathcal{G})$ , donde

- $\Sigma$  es un dominio de planificación.
- $s_0 \in \mathcal{S}$  es el estado inicial.
- $\mathcal{G} \subset \mathcal{S}$  es el conjunto de estados que cumplen el objetivo.

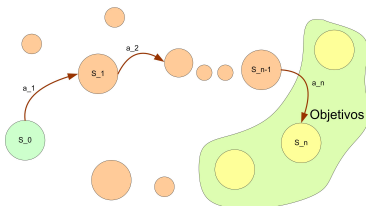
Un *plan* es una sucesión de acciones,  $\pi = (a_1, \dots, a_n)$ .

# Planificación - Introducción

## Representación básica como sistema de transición de estados

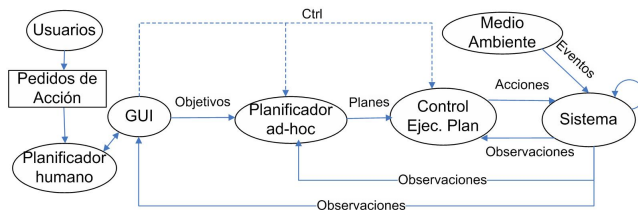
Una **solución** del problema de planificación es un plan  $\pi = (a_1, \dots, a_n)$  que cumple lo siguiente.

- Existe una sucesión de estados  $(s_0, \dots, s_n)$  tales que  $\gamma(s_{i-1}, a_i) = s_i$  para  $i = 1, \dots, n$ .
- $s_0$  es el estado inicial.
- $s_n \in \mathcal{G}$ , es decir,  $s_n$  cumple el objetivo.



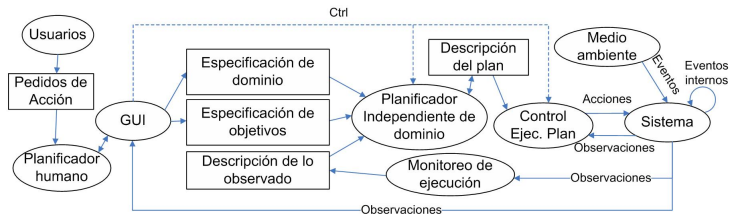
# Planificación - Introducción

## Planificación dependiente del dominio



- Más eficiente (se pueden implementar heurísticas dependientes del dominio).
- Más fácil de implementar.
- Poca flexibilidad al cambio de objetivos y dominio.

## Planificación *in*dependiente del dominio.



- Puede ser re-utilizado.
- Fácilmente adaptable a cambios en el dominio.
- Menos eficiente.

## **Presentación**

## **Planificación**

**Introducción**

**Representación y algoritmos**

## **Scheduling**

**Descripción de los problemas**

**Algunas técnicas utilizadas**

## **Operaciones de Misiones Satelitales**

# Planificación - Representación y Algoritmos

Cada sistema (*framework*) o paradigma de planificación se caracteriza por lo siguiente.

- Cómo se describen o codifican los estados, las acciones y los objetivos.
- El nivel de expresividad que ofrece: restricciones sobre el dominio (mundo cerrado, determinístico), características del dominio (recursos, tiempo, concurrencia explícita, incertidumbre), complejidad de objetivos (*extended goals*), etc.
- Los algoritmos que implementa.



## Situation Calculus [McCarthy & Hayes1969]

- Basado en lógica de primer orden con el agregado de algunas proposiciones especiales.
- Idea: codificar el problema de planificación como un teorema a demostrar.
- La transición entre estados se deduce de axiomas específicos para tal fin.
- Orientado a algoritmos probadores de teoremas.
- Problema: el problema de contexto o (*frame problem*).
- Alcanzó renombre con GOLOG [Levesque *et al.*1997] y sigue activo actualmente.



## STRIPS [Fikes & Nilsson1971]

- Usa predicados para describir los estados.
- Se especifica cada operador (acción) por medio de sus precondiciones y sus efectos, que son también predicados.
- Idea: separar el modelado de los estados de las acciones que pueden actuar en ellos.
- Los estados son definidos implícitamente por las proposiciones que son ciertas en ellos.
- Los operadores tienen parámetros cuya instanciación genera las acciones.
- Orientado a algoritmos de búsqueda.

Estandar de Lenguaje de Planificación (*Planning Domain Definition Language, PDDL* [McDermott *et al.*1998, Fox & Long2003] ).

- Es un lenguaje para codificar un problema al estilo STRIPS.
- Muchas extensiones (tiempo explícito, efectos condicionales, cuantificación en efectos, etc).
- Se usa en la competencia internacional de planificación. automática.

## Ejemplo de especificación de operador en PDDL (Dominio Logistics).

```
(define (domain logistics-strips) ...  
(:action LOAD-TRUCK  
:parameters (?obj - object ?truck - vehicle ?loc -  
location)  
:precondition (and (at ?truck ?loc) (at ?obj ?loc))  
:effect (and (not (at ?obj ?loc)) (in ?obj ?truck))  
))
```

- Siguiendo este paradigma, los algoritmos de planificación se reducen esencialmente a algoritmos de búsqueda (en grafos):
  - amplitud,
  - profundidad,
  - mejor-primero,
  - escalada,
  - $A^*$ , etc.
- *Punto clave*: construcción de heurísticas independientes del dominio.

## Partial Order Planning (POP) [Weld1994]

- La especificación del problema es similar a STRIPS.
- La búsqueda es en el espacio de planes en lugar de en el espacio de estados.
- La búsqueda en el espacio de planes permite algoritmos más sofisticados basados en agregar restricciones y links causales entre acciones.
- Se le está volviendo a prestar atención para dominios temporales [Coles *et al.*2010]

## Enfoques más modernos

- **GRAPHPLAN** [Blum & Furst1997]: Se basa en el uso de grafos auxiliares que permiten identificar las relaciones entre las proposiciones y las acciones. Básicamente, instancia todos los operadores y arma un grafo que conecta capas de acciones con capas de predicados.
- **SATPLAN** [Kautz & Selman1992]: se codifica y resuelve el problema como un problema de satisfacibilidad lógica.
- **HSP** [Bonet & Geffner2001]: construye heurísticas en base a relajar las acciones no considerando los efectos negativos.
- **Metric-FF** [Hoffmann2003]: transforma esas heurísticas para ser capaz de manejar *fluents* (variables numéricas en los estados).
- **HTN** (*Hierarchical Task Network*) [Erol, Hendler, & Nau1994] permite especificar macros de acciones, reduciendo los tiempos de búsqueda.
- Basado en **CSP** (Constraint Satisfaction Problem) [Cesta, Fratini, & Oddi2004]: se codifica y resuelve el problema como un problema de satisfacción de restricciones (el modelado en este caso es muy distinto).
- Muchos más...

## Planificación

Introducción

Representación y algoritmos

## Scheduling

Descripción de los problemas

Algunas técnicas utilizadas

## Operaciones de Misiones Satelitales

Familia de problemas de optimización combinatoria que consiste en tomar decisiones sobre la asignación de tareas, tiempos de ejecución y recursos.

Suelen ser mucho más específicos que los problemas de planificación (en general el modelado del problema ya viene dado).



Existen varias extensas familias de este tipo de problemas, por ejemplo [Pinedo1998]:

- ➊ Flow Shop Scheduling: cada trabajo pasa una vez por cada máquina, todas en el mismo orden.
- ➋ Open Shop Scheduling: cada trabajo pasa una vez por cada máquina, no importa el orden.
- ➌ Job Shop Scheduling: cada trabajo tiene un itinerario asignado.

## Un ejemplo: MINIMUM JOB SHOP SCHEDULING

**Instancia:** un número  $m \in \mathbb{Z}^+$  de procesadores, un conjunto  $\mathcal{J}$  de trabajos. Cada  $j \in \mathcal{J}$  consiste en una secuencia,  $n_j$ , de operaciones  $o_{ij}$ , con  $1 \leq i \leq n_j$ . Para cada operación se asigna un procesador  $p_{ij} \in [1, \dots, m]$  y un tamaño  $l_{ij} \in \mathbb{N}$ .

**Solución:** un *schedule* para los trabajos, es decir, un conjunto de schedules de un procesador. Los mismos son funciones de la forma  $f_p : \{o_{ij} : p_{ij} = p\} \rightarrow \mathbb{N}$ . Deben satisfacer lo siguiente:

- ❶  $f_p(o_{ij}) > f_p(o_{i'j'}) \Rightarrow f_p(o_{ij}) \geq f_p(o_{i'j'}) + l_{i'j'}$
- ❷  $f_{p'}(o_{(i+1)j}) \geq f_p(o_{ij}) + l_{ij}$ .

**Medida de calidad:** el tiempo que se tarda, en fórmula,  $\max_{j \in \mathcal{J}} f_p(o_{n_j j}) + l_{n_j j}$ .

## Presentación

## Planificación

Introducción

Representación y algoritmos

## Scheduling

Descripción de los problemas

Algunas técnicas utilizadas

## Operaciones de Misiones Satelitales

- Todos los problemas son de gran complejidad.
- Se aplican técnicas de optimización combinatoria, relacionándolos muchas veces con otros problemas.
- Clásicamente se aplicó (dependiendo de cuál es el problema):
  - Programación Lineal,
  - Branch and Bound,
  - Programación dinámica.
- Al Scheduling usualmente aplica técnicas de *Constraint Programming* para poder resolver problemas grandes (e.g., [Policella et al.2007]).

## Presentación

## Planificación

Introducción

Representación y algoritmos

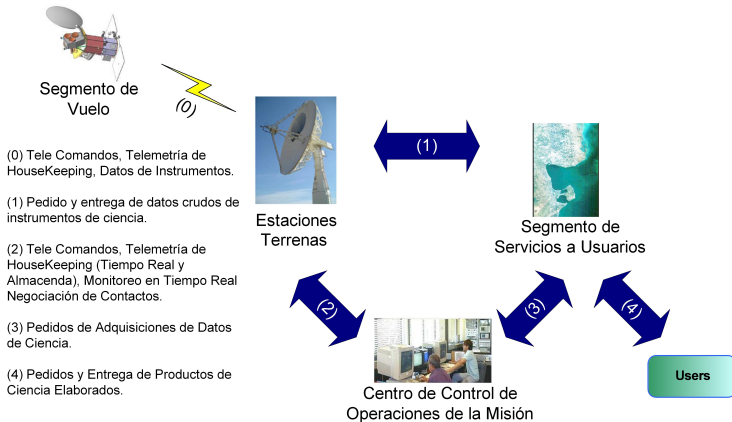
## Scheduling

Descripción de los problemas

Algunas técnicas utilizadas

## Operaciones de Misiones Satelitales

# Operaciones de Misiones Satelitales



Me encuentro estudiando formalismos para unificar o comparar los distintos enfoques.

# Gracias



# Bibliografía I



Blum, A., and Furst, M.

1997.

Fast planning through planning graph analysis.  
90:281–300.



Bonet, B., and Geffner, H.

2001.

Planning as heuristic search.  
*Artif. Intell.* 129(1-2):5–33.



Cesta, A.; Fratini, S.; and Oddi, A.

2004.

Planning with concurrency, time and resources: A CSP-Based approach.  
In Vlahavas, I., and Vrakas, D., eds., *Intelligent Techniques for Planning*. Idea Group Publishing.  
chapter 8, 259–295.



Coles, A. J.; Coles, A. I.; Fox, M.; and Long, D.

2010.

Forward-chaining partial-order planning.  
In *Proceedings of the Twentieth International Conference on Automated Planning and Scheduling (ICAPS-10)*.



Erol, K.; Hendler, J.; and Nau, D.

1994.

HTN Planning: Complexity and Expressivity.  
1123–1128.



Fikes, R., and Nilsson, N.

1971.

STRIPS: A new approach to the application of theorem proving to problem solving.  
2(3-4):189–208.

# Bibliografía II



Fox, M., and Long, D.

2003.

PDDL 2.1: An extension to PDDL for expressing temporal planning domains.

*Journal of Artificial Intelligence Research* 20:61–124.

Special issue on 3rd International Planning Competition.



Ghallab, M.; Nau, D.; and Traverso, P.

2004.

*Automated Planning: Theory & Practice*.

Morgan Kaufmann.



Hoffmann, J.

2003.

The metric-ff planning system: Translating "ignoring delete lists" to numerical state variables.

*JOURNAL OF ARTIFICIAL INTELLIGENCE RESEARCH. SPECIAL ISSUE ON THE 3RD INTERNATIONAL PLANNING COMPETITION* 20.



Kautz, H., and Selman, B.

1992.

Planning as Satisfiability.

*In Proceedings of the 10th European Conference on Artificial Intelligence (ECAI 92)*.



Levesque, H.; Reiter, R.; Lespérance, Y.; Lin, F.; and Scherl, R.

1997.

GOLOG: A logic programming language for dynamics domains.

*Journal of Logic Programming: Special issue on actions* 31(1–3):59–83.

# Bibliografía III



McCarthy, J., and Hayes, P.

1969.

Some philosophical problems from the standpoint of artificial intelligence.

In *Machine Intelligence*, volume 4.

Edinburgh University Press.



McDermott, D.; Ghallab, M.; Howe, A.; Knoblock, C.; Ram, A.; Veloso, M.; Weld, D.; and Wilkins, D.

1998.

PDDL - the planning domain definition language.

Technical Report CVC TR-98-003, DCS TR-1165, Yale Center for Communicational Vision and Control.



Pinedo, M.

1998.

*Scheduling, Theory, Algorithms, and Systems.*

Prentice Hall.



Policella, N.; Cesta, A.; Oddi, A.; and Smith, S. F.

2007.

From precedence constraint posting to partial order schedules: A csp approach to robust scheduling.

*AI Commun.* 20(3):163–180.



Weld, D.

1994.

An Introduction to Least Commitment Planning.

*AI Magazine* 15(4):27–61.