



decsai.ugr.es

Universidad de Granada

Tema 2: Planificación Automática

Curso en Minería de Procesos

Juan Fernández Olivares



**Departamento de Ciencias de la
Computación e Inteligencia Artificial**

En la sesión anterior ...



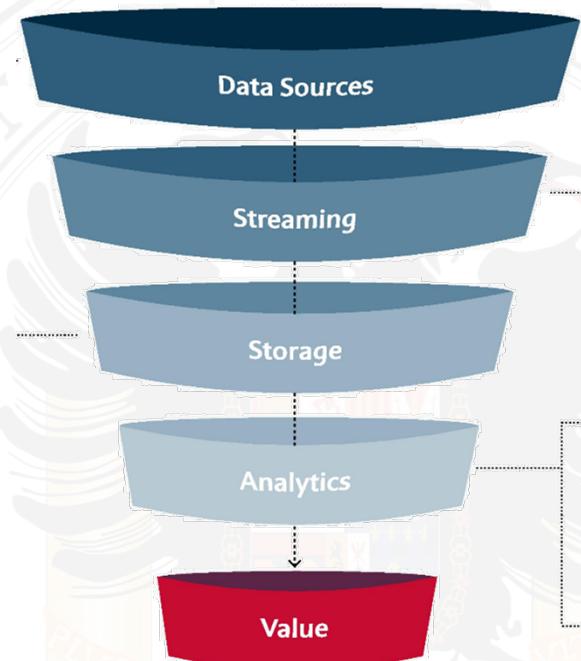
- 1. Analítica de negocio**
- 2. Tipos de analítica de negocio**
- 3. Analítica de negocio vs Minería de Procesos**
- 4. Analítica Prescriptiva: Modelos prescriptivos**
- 5. Planificación automática: tecnología habilitadora para ayuda a la decisión en Minería de Procesos y Analítica Prescriptiva**
 - 1. Planificación clásica**
 - 2. Casos de uso de planificación en Minería de procesos**
 - 3. Planificación HTN: Hierarchical Tasks Networks**
 - 4. Planificación HTN para ayuda a la decisión con modelos normativos: ACM**

Gran parte de los contenidos del Máster en C.D. están orientados a la descripción de técnicas que se engloban en el término Analítica de Negocio (Business Analytics)

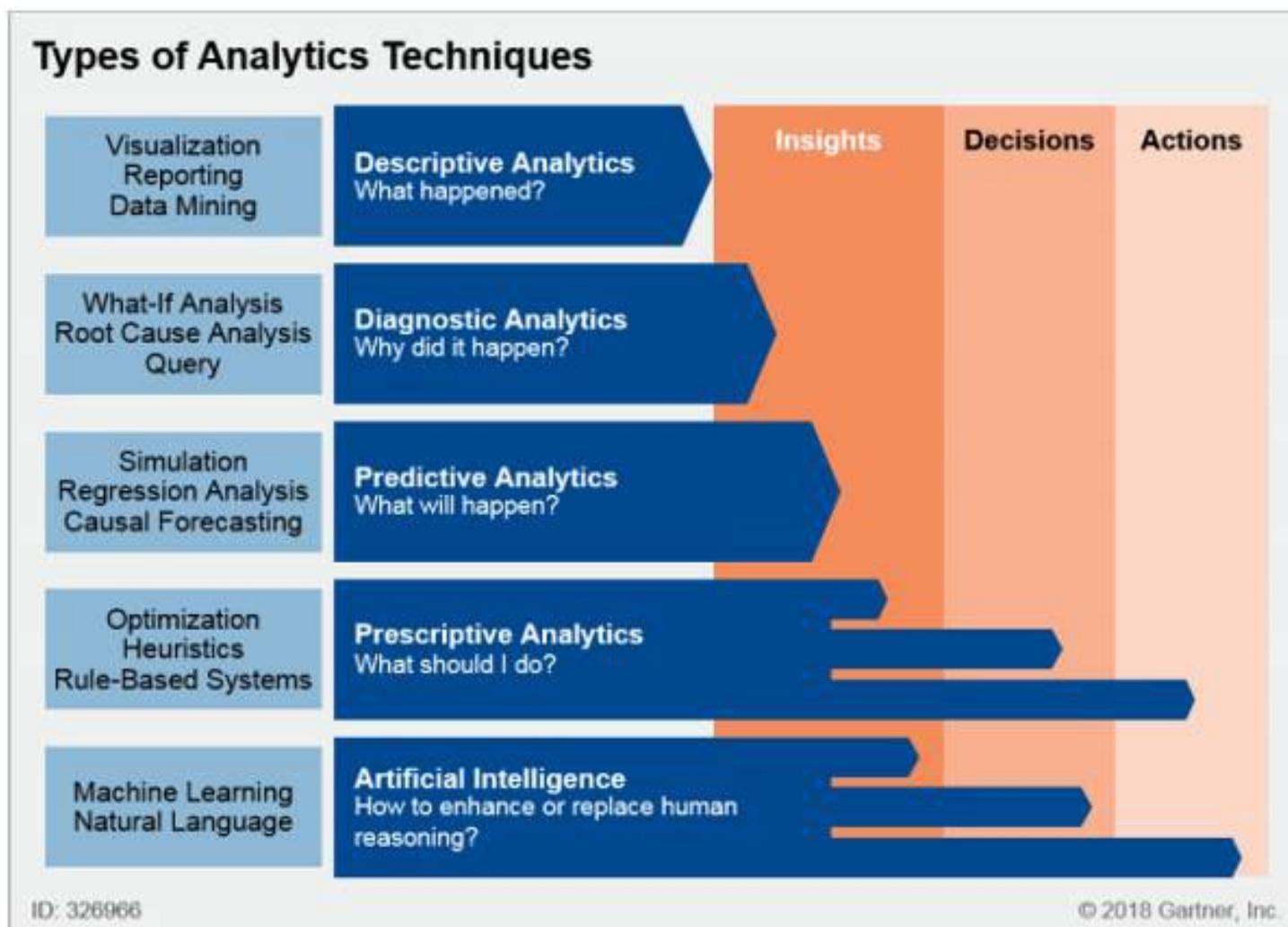
Schniederjans, Marc J., Dara G. Schniederjans, y Christopher M. Starkey. *Business analytics principles, concepts, and applications: what, why, and how*. Pearson Education, 2014.

<https://learning.oreilly.com/library/view/business-analytics-principles/9780133552256/>

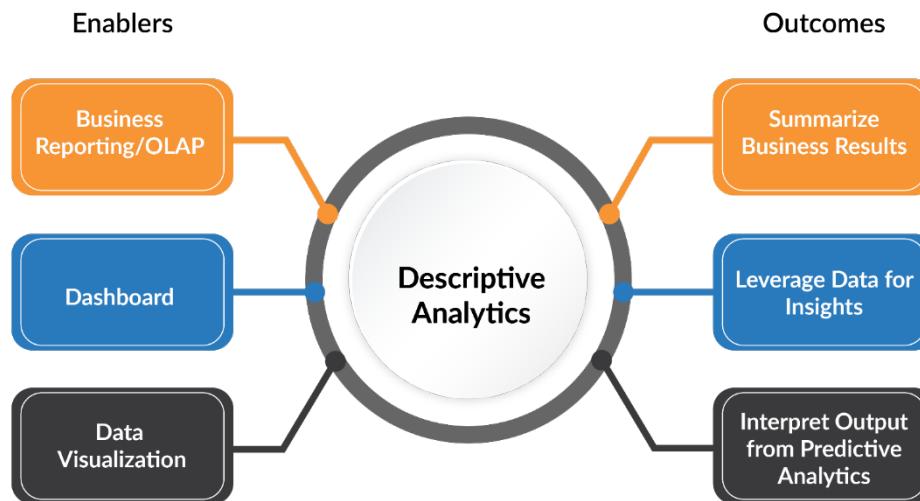
- Analítica =
 - Estadística + Ciencia de la Computación + Bases de Datos+ Aprendizaje Automático + Búsqueda Heurística + ...
 - Explorar, visualizar, descubrir y comunicar patrones y tendencias en los datos.
- Convertir datos en información útil y con valor añadido.
- Analítica se aplica a varias áreas (por ejemplo en la predicción del tiempo) pero aplicada a la automatización de procesos (de negocio) (procesos que se llevan a cabo en una organización) se llama **analítica de negocio**



Descripción según Gartner. <https://www.gartner.com/en>



- **Analítica Descriptiva:** aplicación de técnicas estadísticas básicas que describan lo que hay en los datos. Ej: Un diagrama de barras, cuadros de mando.



Workforce KPI Dashboard Showing Visibility...



This graph/chart is linked to excel, and changes automatically based on data. Just left click on it and select "Edit Data".

- **Analítica Predictiva:** construir modelos predictivos para identificar tendencias o relaciones no directamente observables en análisis descriptivo. Ej: una regresión para mostrar la relación entre edad, peso, ejercicio y la venta de alimentos.

The Predictive Analytics Process

**Pull**

Extract the data
from where it lives.

**Prepare**

Clean, refine, and
prepare it.

**Pick**

Identify what to
predict.

**Predict**

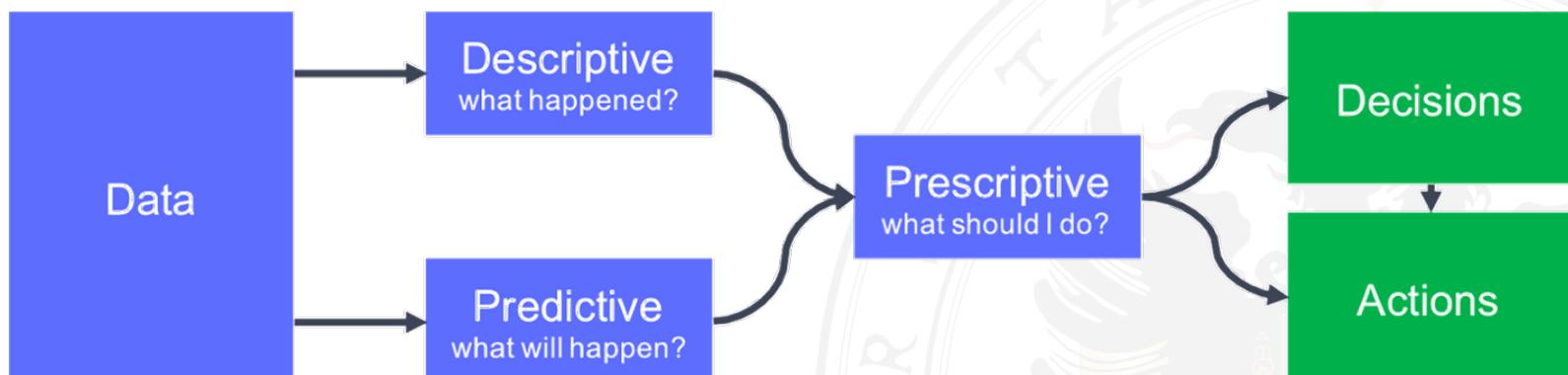
Create the
prediction.

**Plan**

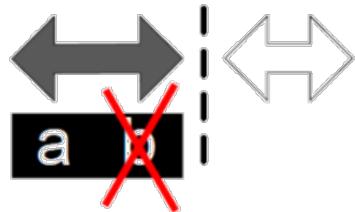
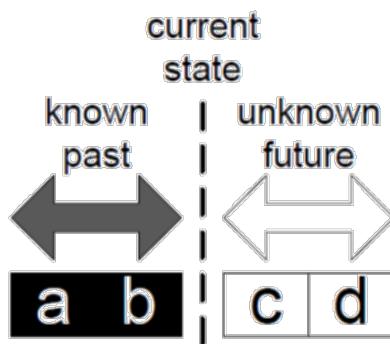
Develop a plan of
action.

- **Analítica Prescriptiva:** determinar y proyectar el uso de las alternativas de acción y recursos disponibles. Ej: un depto. tiene un presupuesto limitado para hacer publicidad, ¿cómo asignar presupuesto de forma óptima para varios medios publicitarios?

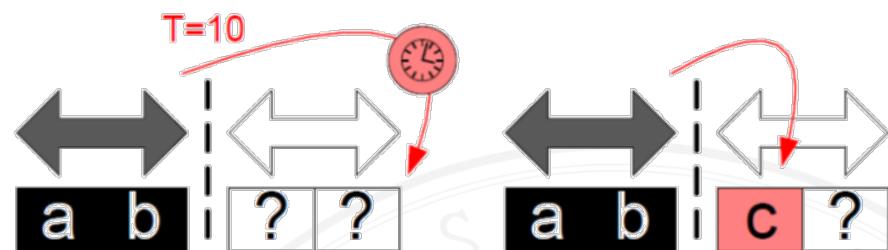
Using Descriptive, Predictive and Prescriptive Analytics to Make Decisions / Take Actions



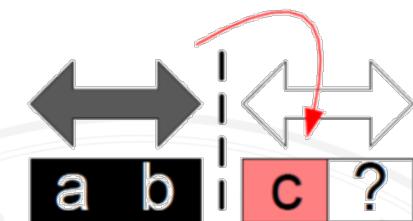
Observar el paralelismo entre los tipos de analítica y las operaciones de ayuda a la decisión en el entorno de Process Mining



detect: b does not fit the model (not allowed, too late, etc.)



predict: some prediction is made about the future (e.g. completion date or outcome)



recommend: based on past experiences c is recommended (e.g., to minimize costs)

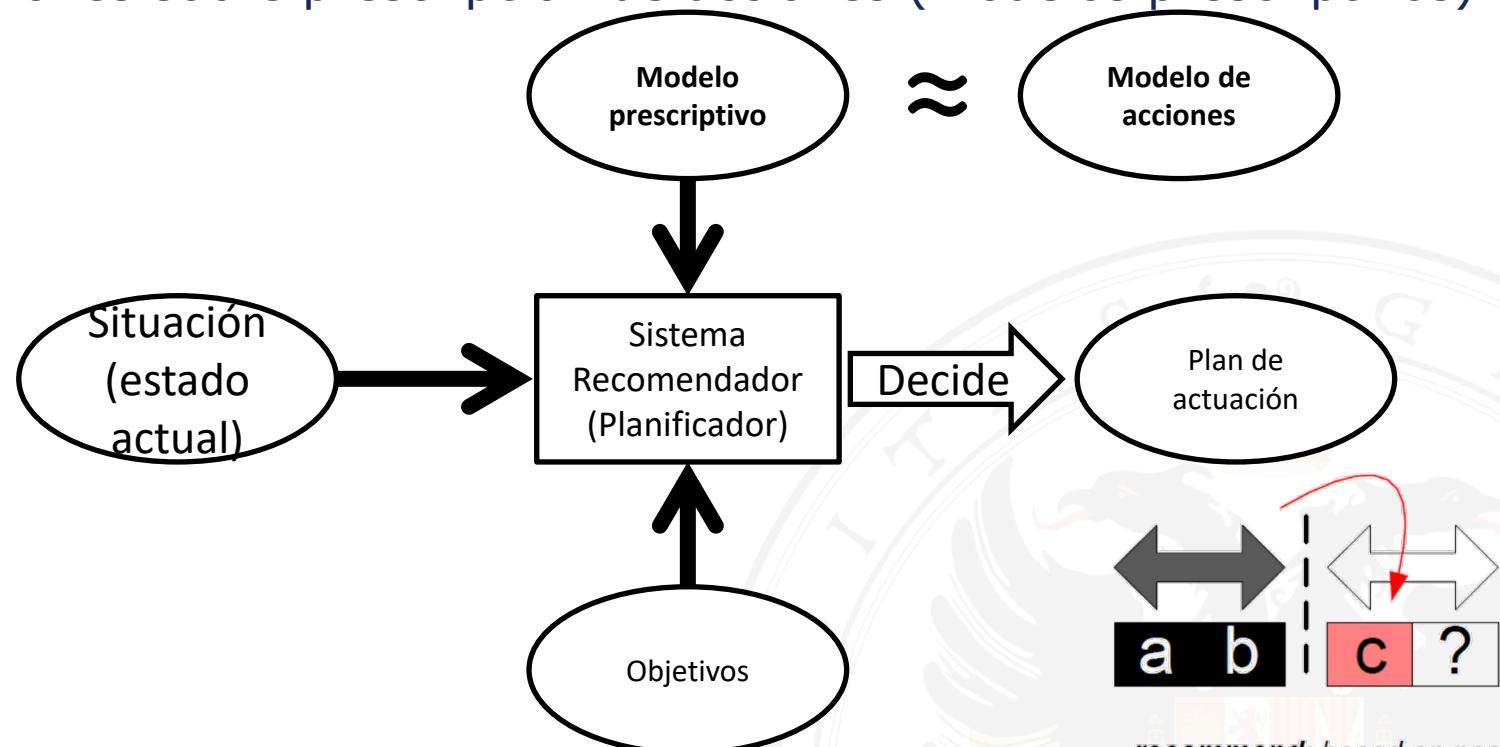
Analítica
descriptiva

Analítica
predictiva

Analítica
prescriptiva

Si con la **analítica predictiva** buscamos construir un **modelo predictivo** para dar una respuesta a **qué puede ocurrir** en el futuro, con la **prescriptiva** buscamos construir un **modelo prescriptivo** que nos de respuesta a **cómo alcanzar una situación determinada (objetivo)** en el futuro.

En este tema nos vamos a centrar en la planificación como una técnica de Inteligencia Artificial para definir y usar modelos de acciones que sirvan como modelos para ayudar a la toma de decisiones sobre prescripción de acciones (modelos prescriptivos).



DECISIÓN = SECUENCIA DE ACTIVIDADES

SOPORTE A LA DECISIÓN = RECOMENDACIÓN/PRESCRIPCIÓN DE ACTIVIDADES

recommend: based on past experiences c is recommended (e.g., to minimize costs)

- Preguntas:

- ¿Cómo podemos **crear un modelo** que sirva para recomendar acciones (modelo prescriptivo como **un modelo de acciones**) orientadas a objetivos?
- ¿Cómo se pueden **sintetizar automáticamente** procesos que dependen de una situación , y de objetivos, a partir de estos modelos ?
- ¿Cómo pueden **modificarse los procesos automáticamente** respondiendo a **cambios de situaciones** o a eventos externos, ajustándose a la nueva situación?
- ¿Cómo podemos **descubrir/aprender modelos prescriptivos?** (siguiente tema)



- Preguntas:

- ¿Cómo podemos **crear un modelo** que sirva para recomendar acciones (modelo prescriptivo como **un modelo de acciones**) orientadas a objetivos?
- ¿Cómo se pueden **sintetizar automáticamente** procesos que dependen de una situación , y de objetivos, a partir de estos modelos ?
- ¿Cómo pueden **modificarse los procesos automáticamente** respondiendo a **cambios de situaciones** o a eventos externos, ajustándose a la nueva situación?
- ¿Cómo podemos **descubrir/aprender modelos prescriptivos?** (siguiente tema)



Utilizando técnicas de
Inteligencia Artificial

- Respuestas:
 - Vienen de la mano de la **Planificación Automática** un área de la **IA** que se dedica a estudiar **cómo se debe actuar para alcanzar objetivos.**



Nau, Dana, Malik Ghallab, and Paolo Traverso. *Automated Planning: Theory & Practice: Theory and Practice*. Amsterdam ; Boston: Morgan Kaufman Publ Inc, 2004.

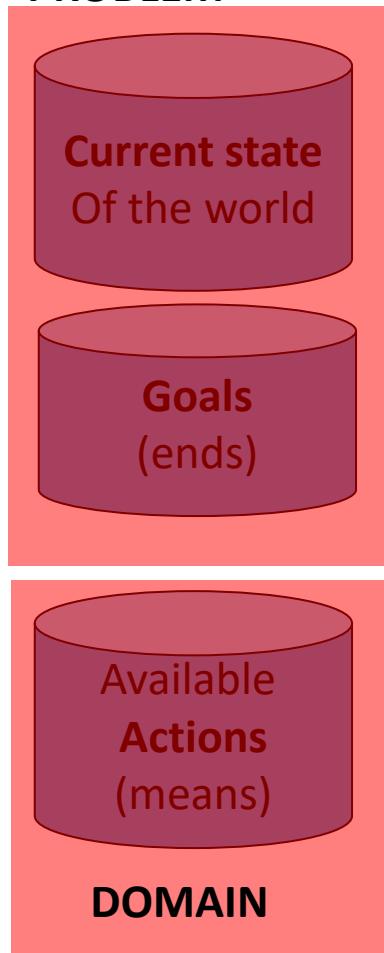
<http://projects.laas.fr/planning/aptp/index.html>

Automated Planning and Acting © Malik Ghallab, Dana Nau and Paolo Traverso. Published by Cambridge University Press, ISBN: 9781107037274, August 2016.

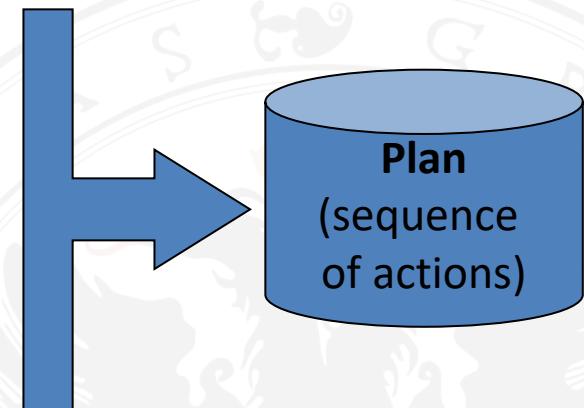
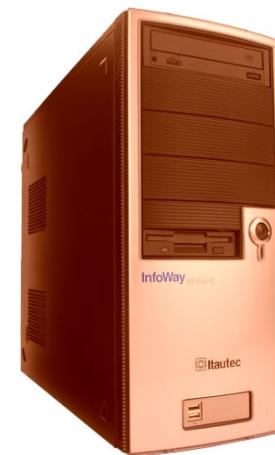
<http://projects.laas.fr/planning/>

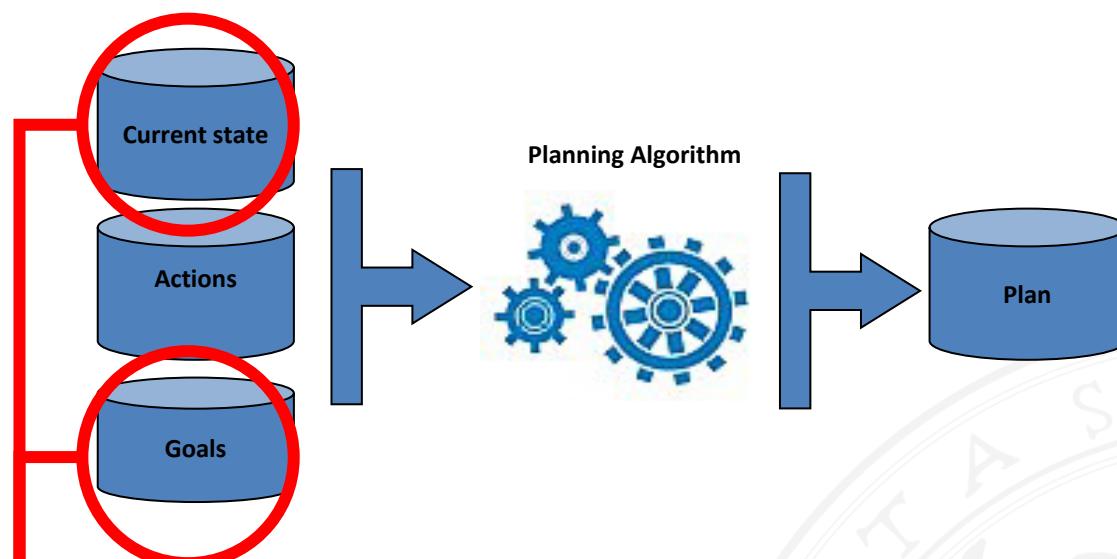
La Planificación Automática (Automated Planning) es una disciplina de la IA que estudia la **generación de planes** (procesos) a partir de un **modelo de acciones**, de unos **objetivos** y una **situación** de partida.

PROBLEM



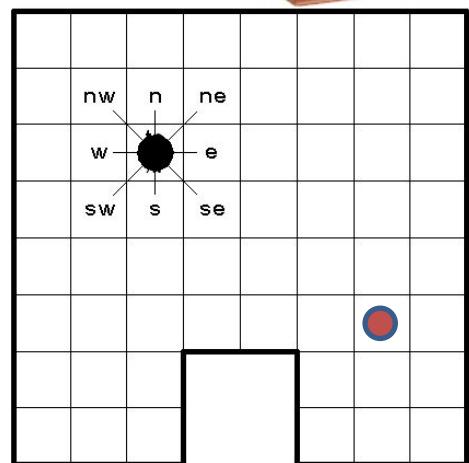
Planning Algorithm



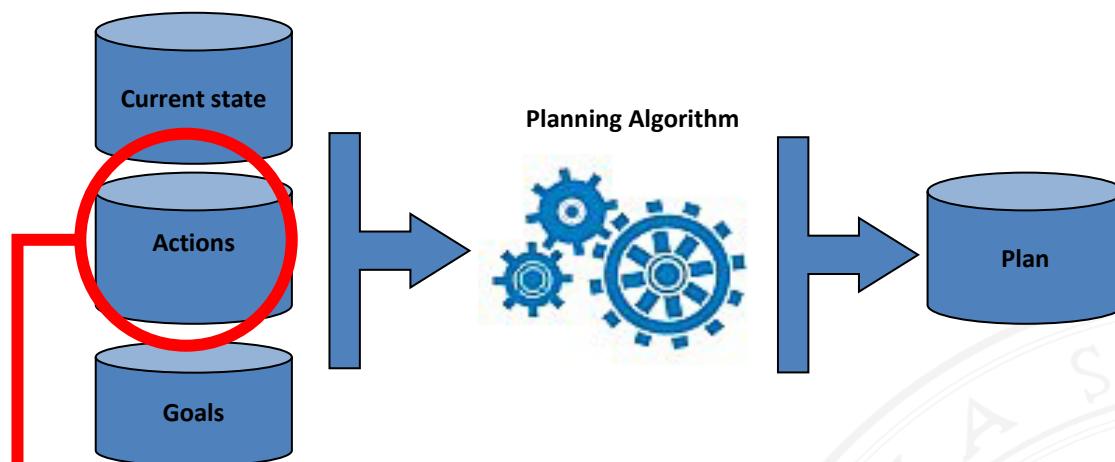


I. Formal representation languages

- Predicate Logic (or a subset)
 - I am at (3,6) and I want to be at (7,3)
 - Current State -> at(3,5)
 - Goal -> at(7,3)
- PROBLEM**



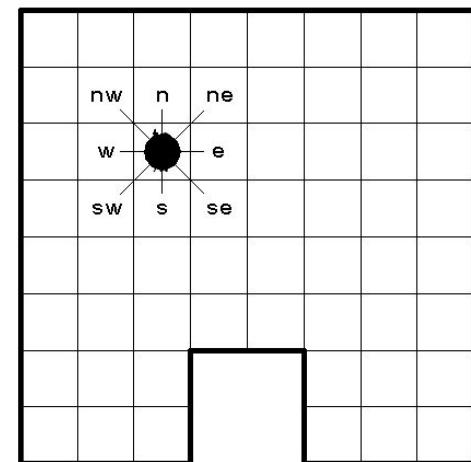
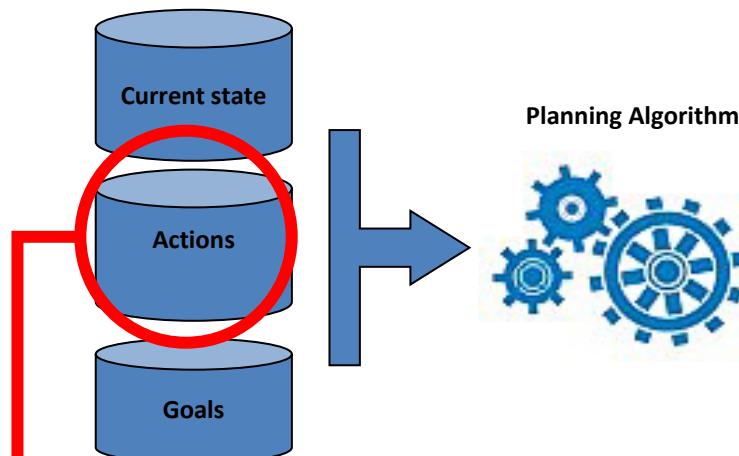
Planificación automática

**II. Models of action: Representing change in the environment**

- Predicate logic
- Structured models

DOMAIN





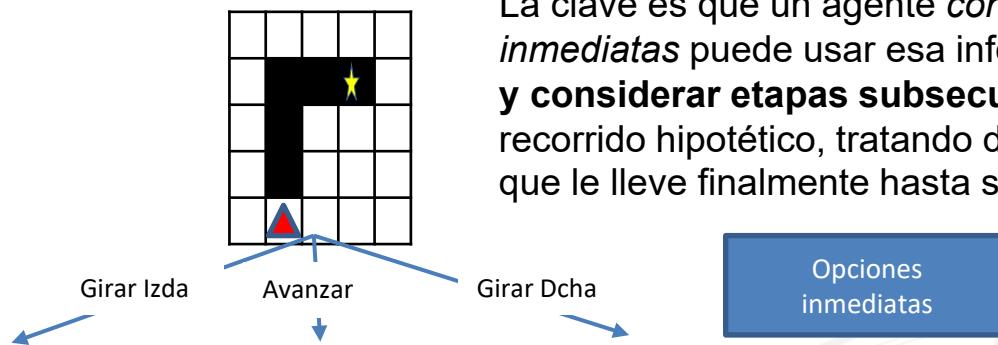
II. Models of action: Representing change in the environment

- Predicate logic
- Structured models

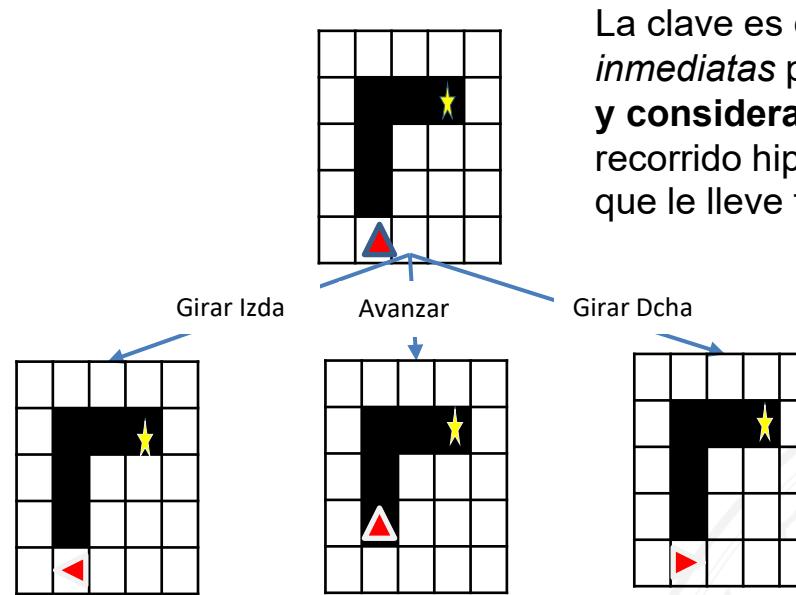
DOMAIN

action: `move_north(robot)`
 Precondition: `at(robot, x, y)`
 Effect: `at(robot, x, y+1)`
`not(at(robot, x, y))`

Es **esencial** conocer las consecuencias de las acciones para saber cómo evoluciona el mundo y tomar decisiones bien informadas



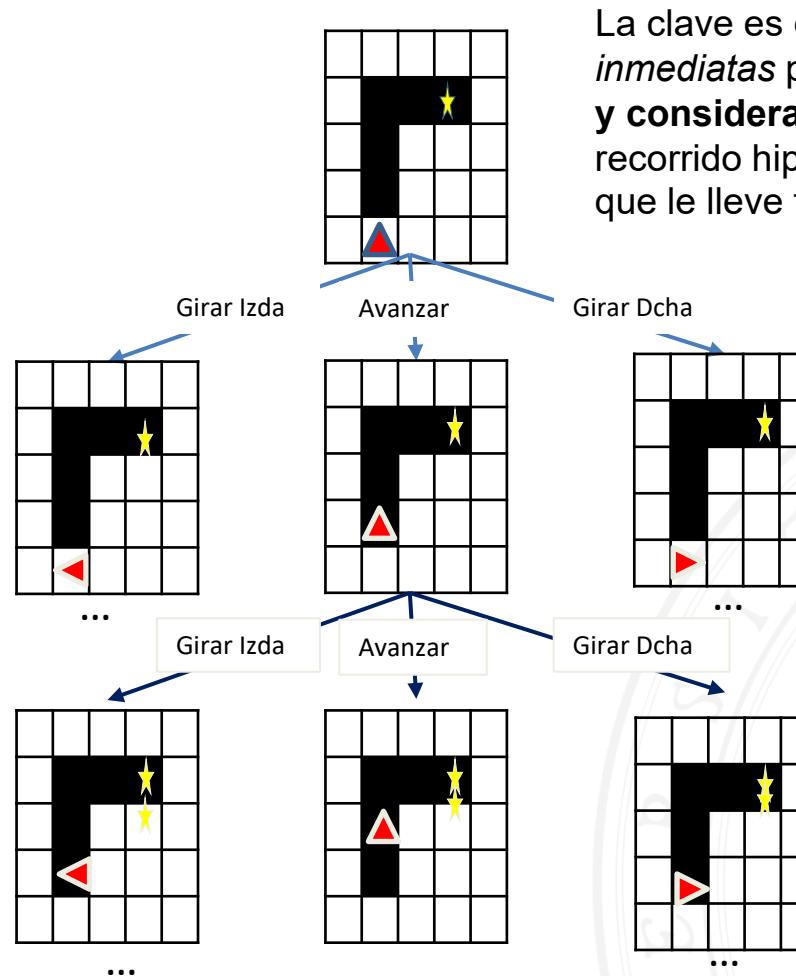
La clave es que un agente *con varias opciones inmediatas* puede usar esa información para **proyectar y considerar etapas subsecuentes** sobre un recorrido hipotético, tratando de encontrar un camino que le lleve finalmente hasta su objetivo.



La clave es que un agente *con varias opciones inmediatas* puede usar esa información para **proyectar y considerar etapas subsecuentes** sobre un recorrido hipotético, tratando de encontrar un camino que le lleve finalmente hasta su objetivo.

Opciones inmediatas

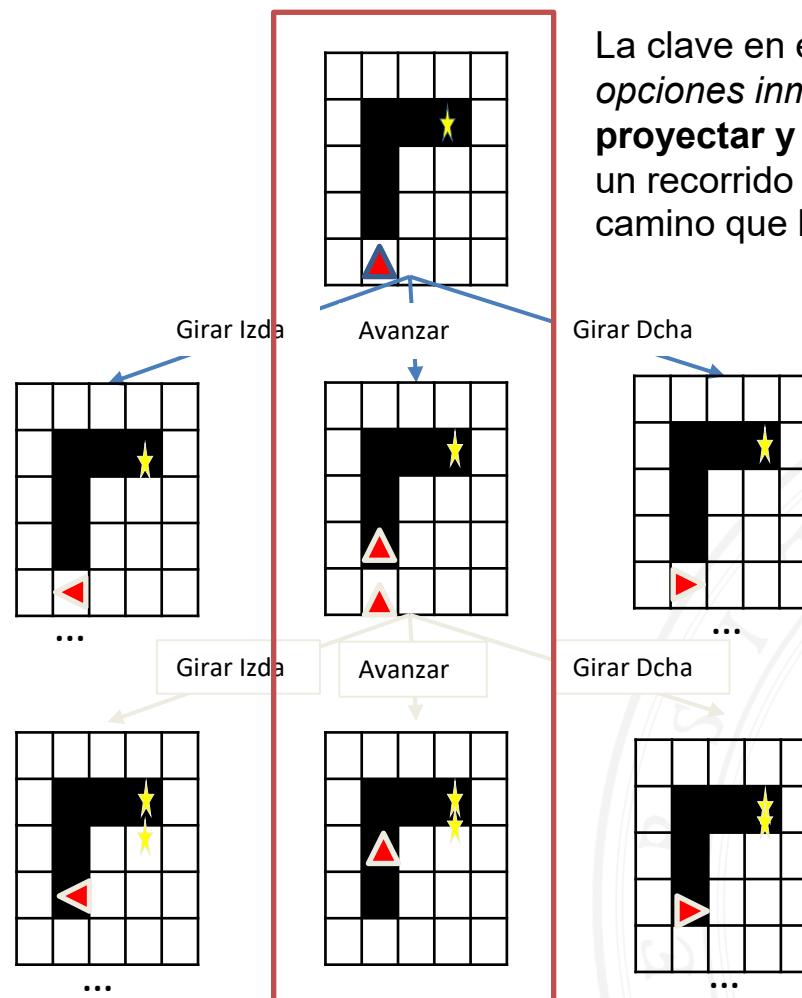
Proyectar hacia el futuro las consecuencias de sus acciones



La clave es que un agente *con varias opciones inmediatas* puede usar esa información para **proyectar y considerar etapas subsecuentes** sobre un recorrido hipotético, tratando de encontrar un camino que le lleve finalmente hasta su objetivo.

Opciones inmediatas

Proyectar hacia el futuro las consecuencias de sus acciones

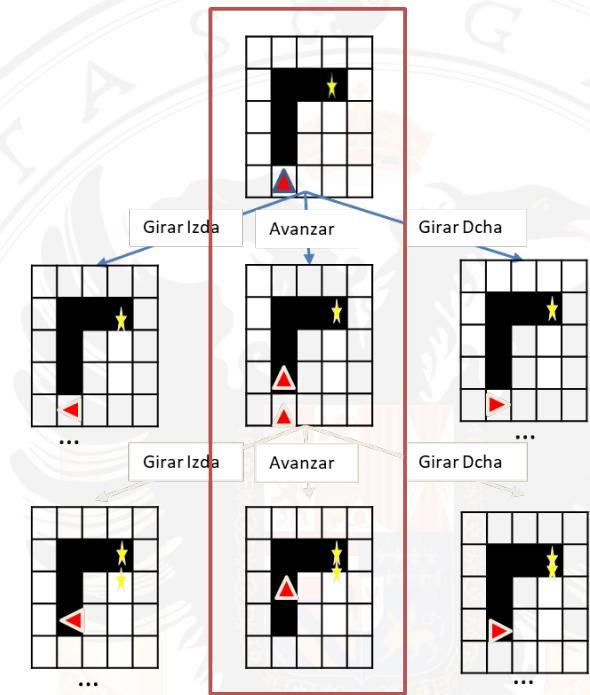
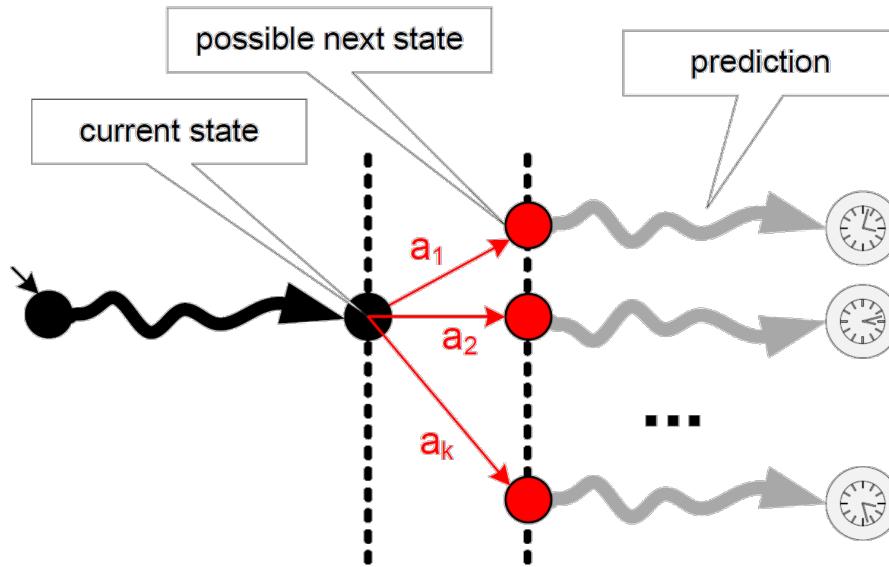


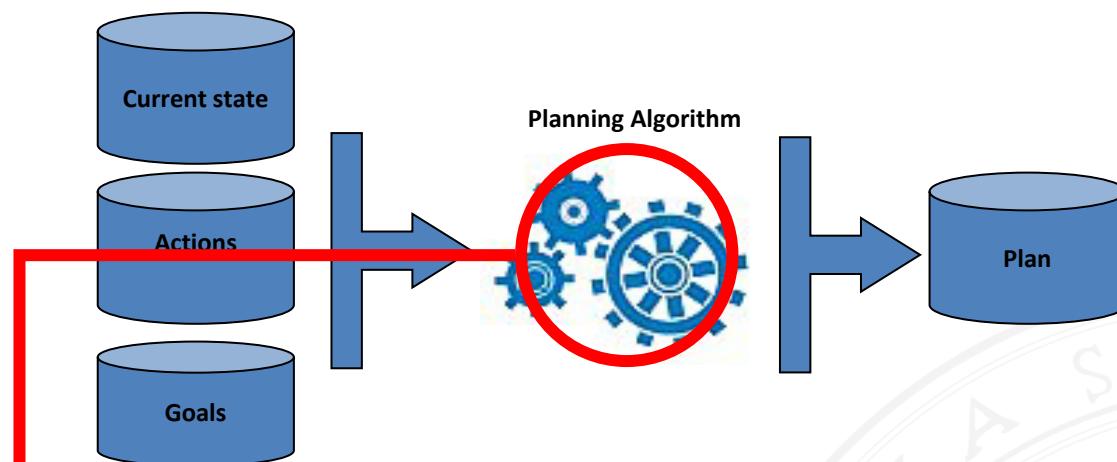
La clave en este tema es que el agente *con varias opciones inmediatas* puede usar esa información para **proyectar y considerar etapas subsecuentes** sobre un recorrido hipotético, tratando de encontrar un camino que le lleve finalmente hasta su objetivo.

Opciones inmediatas

Proyectar hacia el futuro las consecuencias de sus acciones

Encontrar un camino

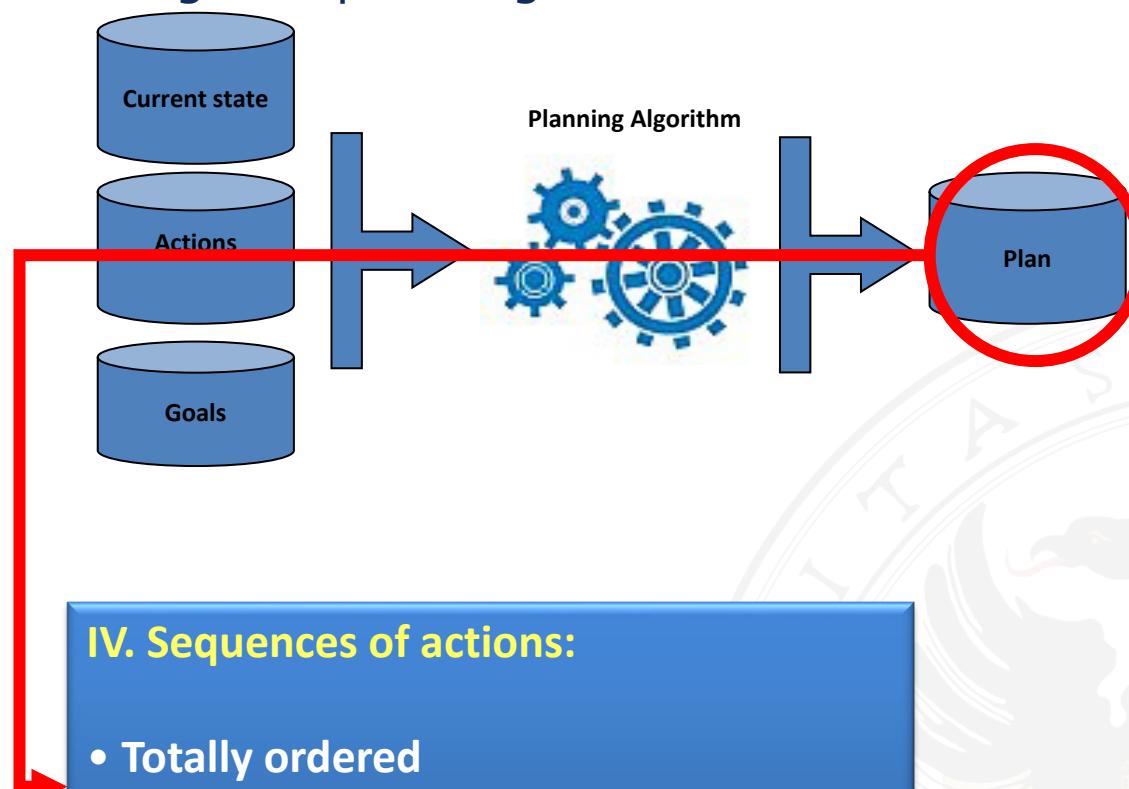




III. Search algorithms

- Blind search
- Heuristic search
- Forward search
- Backwards search

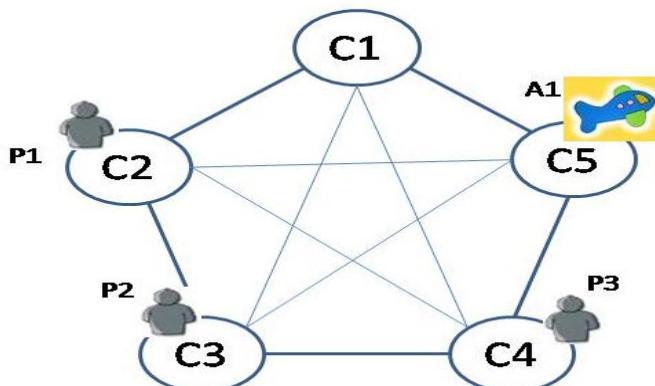
Artificial Intelligence planning



- Planificación automática:

- Técnica de Inteligencia Artificial para la resolución de problemas cuya solución es una secuencia de actividades.
- que deben ejecutarse para alcanzar un objetivo desde una situación de partida.
- Podemos ver la planificación automática desde dos puntos de vista que veremos a continuación

- Visión 1: Planificación automática como **Resolución de problemas**:
 - Estado inicial: (ver figura)
 - Estado objetivo: (todas las personas en C1)
 - Operadores (Dominio de planificación):
 - volar(a,x,y), embarcar (a,p,x),desembarcar (a,p,x)
 - Proceso de búsqueda
 - Solución: secuencia de actividades -> plan
 - Volar(a1,c5,c2), Embarcar(a1,p1,c2),Volar(a1,c2,c3),....
 - Genera una secuencia de actividades, pero no una cualquiera, sino una que define el comportamiento de un agente (físico o software) para alcanzar un objetivo.

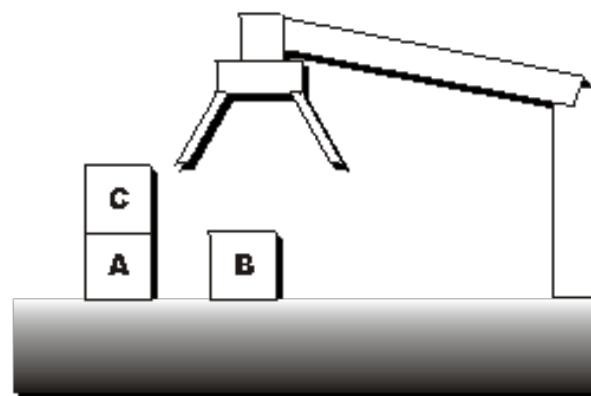


Ejemplo:

- Transporte aéreo entre ciudades
- Personas, ciudades, aviones
- En concreto: 5 ciudades, 1 avión, 3 personas
- ¿Qué acciones deben ejecutarse para transportar todas las personas a C1?
- ¿Qué plan de acciones tiene el mínimo consumo de combustible?

- Visión 2: Planificación automática para generar procesos
 - Un **planificador es un Generador de procesos.**
 - Secuencia de actividades, un plan, **es un proceso (o instancia de proceso).**
 - Entonces la planificación automática puede verse como la generación de
 - procesos orientados a objetivos (planes)
 - a partir de un modelo de proceso (dominio de planificación).

Problema clásico: Blocks world



Operadores

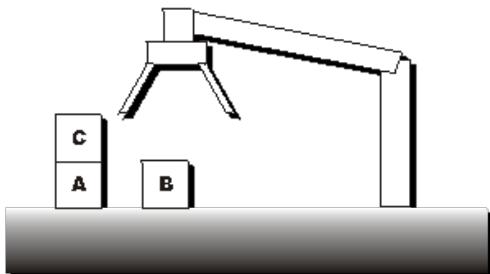
Pickup(X)

Putdown(X)

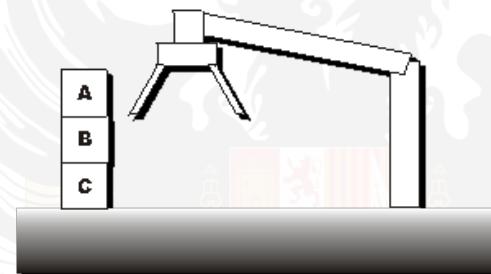
Stack(X,Y)

Unstack(X,Y)

Estado Inicial



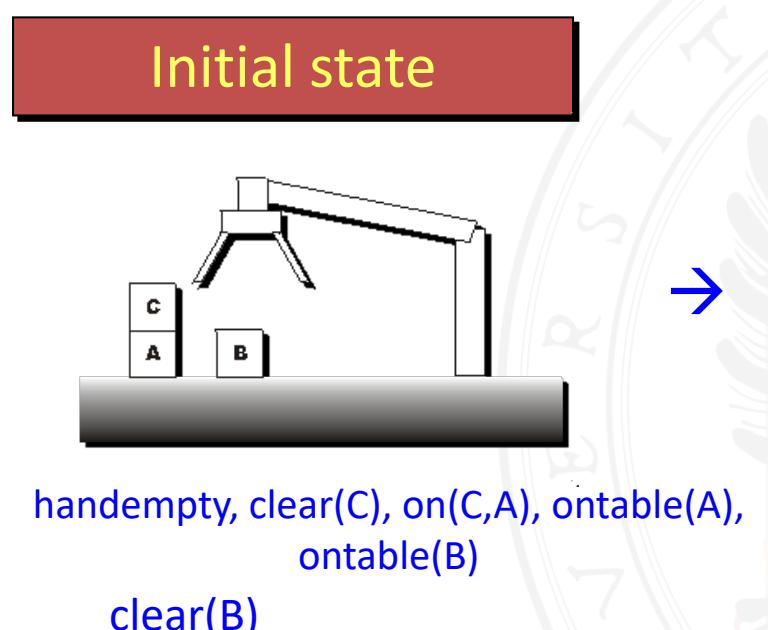
Goal



Pickup(C), Putdown(C), Pickup(B), Stack(B,C), Pickup(A), Stack(A,B)

Plan

- Lenguaje de representación
 - Representación de estados:
 - Un conjunto de predicados instanciados describiendo una situación.
 - Asunción del mundo cerrado
 - Cualquier hecho no presente en el estado inicial o intermedios se supone FALSO



→ $\neg \text{clear}(B)$

- **Modelo de acciones:**

- A structured model based on predicate logic. Every action is described by three components:
 - **Precondition list:** Facts that MUST BE true prior to the execution of the action.
 - **Delete list:** Facts that are no longer true due to the execution of the action.
 - **Add list:** Facts that become true after the execution of the action

PDDL: un lenguaje para representar modelos de acciones

handempty
holding(X)
clear(X)
on(X,Y)
ontable(X)

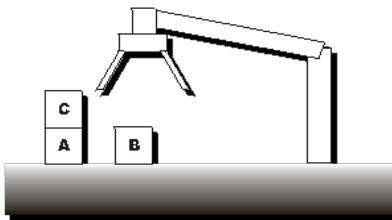
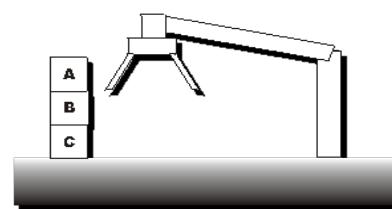
```
(:action unstack
:parameters (?x ?y – block)
:precondition (and (handempty)(clear ?x)(on ?x ?y))
:effect (and (not (handempty)) (not (clear ?x))
             (not (on ?x ?y))(holding ?x) (clear ?y)))
```

Unstack(X,Y)

Precondition:	handempty, clear(X), on(X,Y)
Delete:	handempty, clear(X), on(X,Y)
Add:	holding(X), clear(Y)

PDDL : Planning Domain Description Language
(Long & Fox, 2003, Edelkamp & Hoffman 2004)

https://en.wikipedia.org/wiki/Planning_Domain_Definition_Language

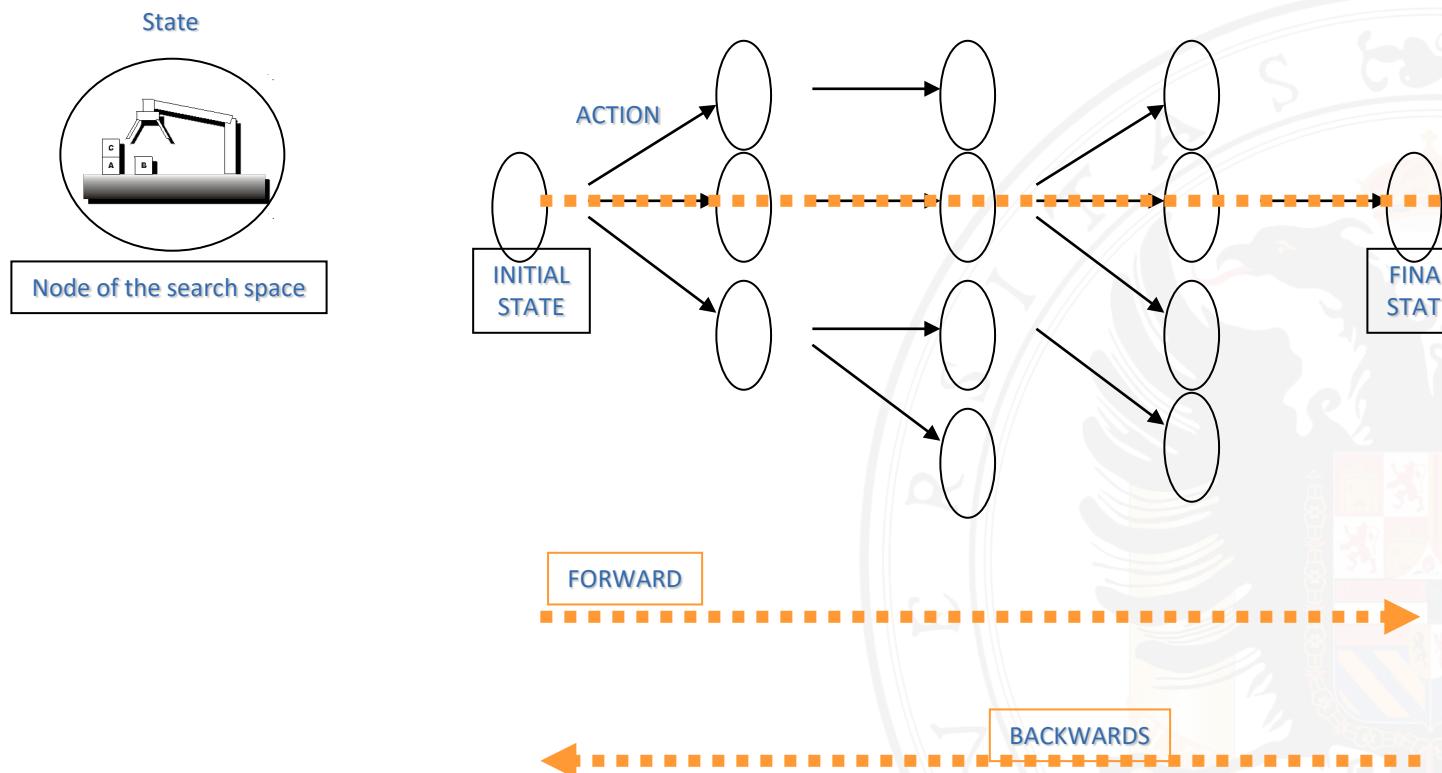
Estado Inicial**Goal****Problema PDDL**

```
(define (problem BLOCKS-4-0)
  (:domain BLOCKS)
  (:objects B A C - block)
  (:INIT
    (CLEAR C)
    (CLEAR B)
    (ONTABLE A)
    (ONTABLE B)
    (ON C A)
    (HANDEMPTY) )
  (:goal (AND (ON A B) (ON B C)))) )
```

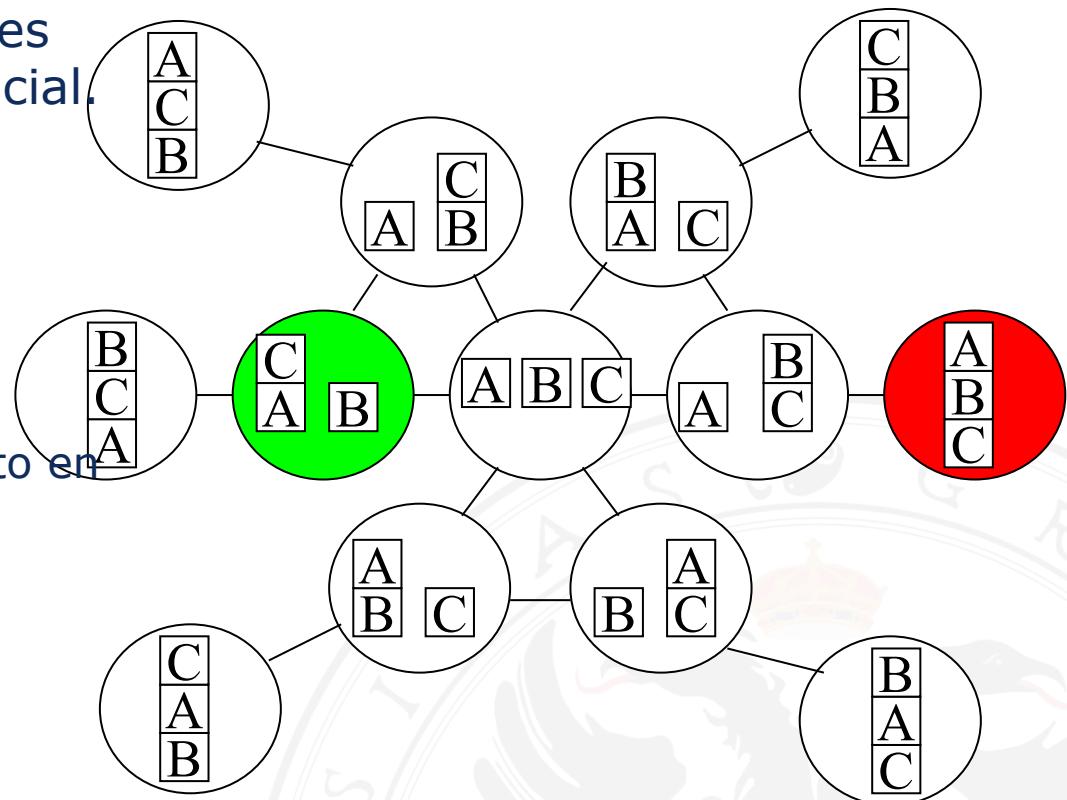
Dominio PDDL**Planificador**

pickup(C),
putdown(C),
Pickup(B)
stack(B,C),
Pickup(A)
stack(A,B)

- El proceso de búsqueda de una solución:
 - Búsqueda en un espacio de estados
 - *operadores espacio búsqueda == acciones*
 - *generación de estados sucesores mediante aplicación de acciones*
 - *se detiene cuando encontramos el estado objetivo y*
 - *devuelve el camino desde el estado inicial hasta el objetivo.*

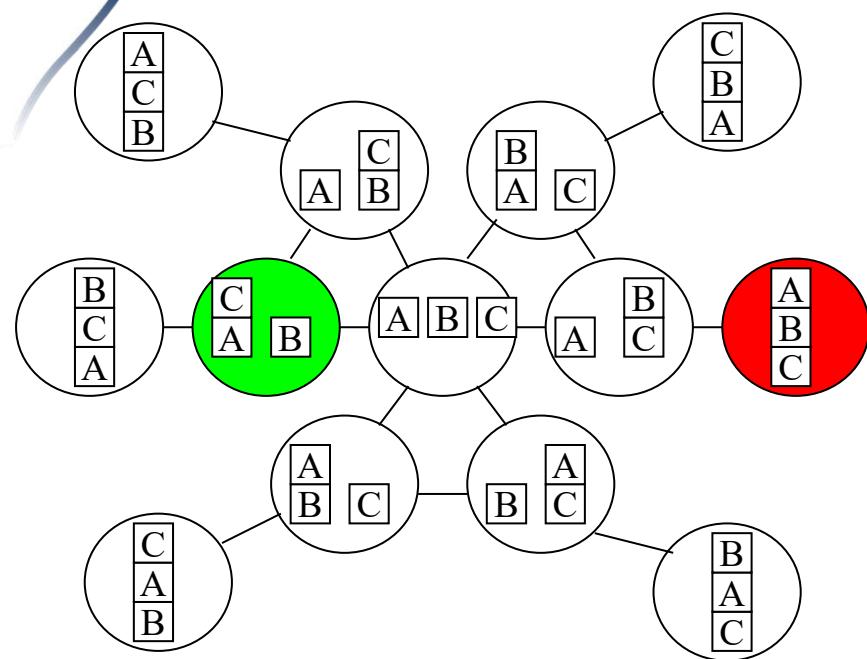


- El espacio de búsqueda es de complejidad exponencial.
- Por eso se empezaron usando técnicas de búsqueda regresiva.
- Técnicas de búsqueda heurística.
 - Las que se han impuesto en los últimos años.

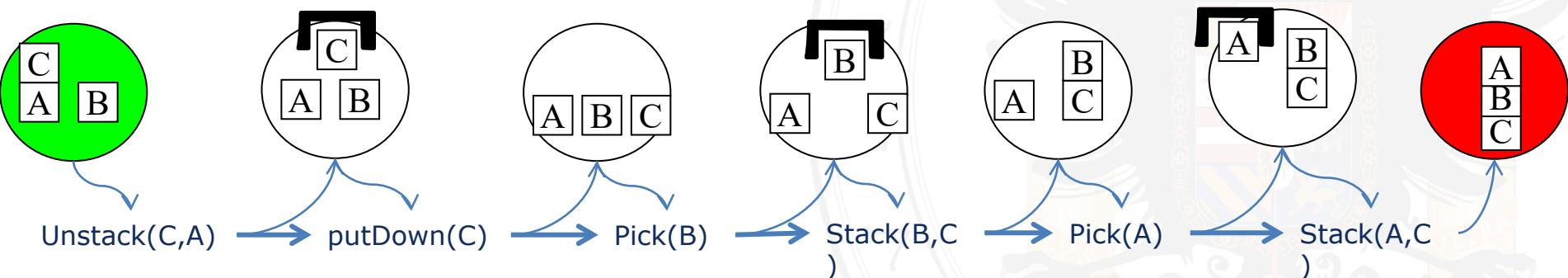


Hoffmann, Jörg, and Bernhard Nebel. "The FF Planning System: Fast Plan Generation through Heuristic Search." *Journal of Artificial Intelligence Research*, 2001, 253–302.

Helmer, Malte. "The Fast Downward Planning System." *J. Artif. Intell. Res.(JAIR)* 26 (2006): 191–246.

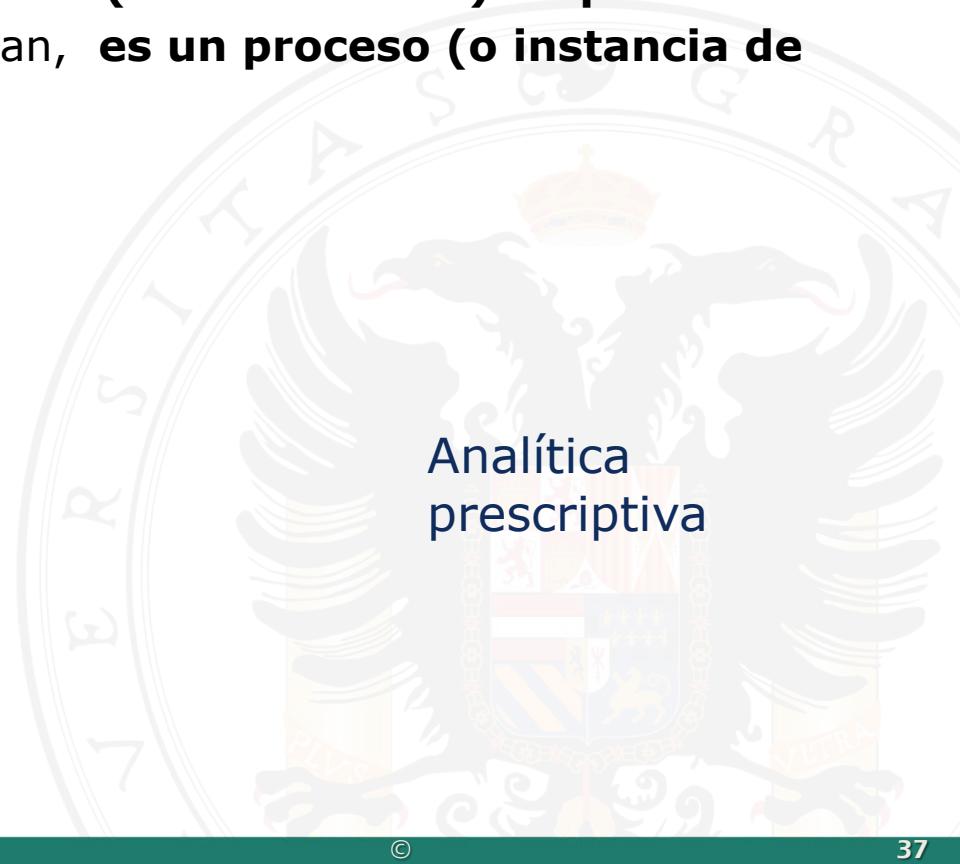


- El plan obtenido es una secuencia totalmente ordenada (en este ejemplo)
- El plan es el camino entre el estado inicial y el final.
- Secuencia de pasos para alcanzar el objetivo.
- La planificación se puede ver como la síntesis de procesos orientados a objetivos.



- 1. conocimos el concepto de analítica prescriptiva,**
 1. la fase de la analítica de negocio que va después de la analítica descriptiva y predictiva y que permite dar respuesta a la pregunta de ¿qué debo hacer? para alcanzar un objetivo determinado a partir de la situación actual.
- 2. relación entre analítica prescriptiva y la parte de minería de procesos dedicada a la toma de decisiones online.**
- 3. Planificación automática:**
 1. tanto desde la minería de procesos como en la analítica de negocio, hay que dar respuesta o ayudar a la toma de decisiones sobre cómo actuar ante situaciones para alcanzar objetivos,
- 4. Planificador recibe como entrada estado actual, objetivo y dominio**
 1. PDDL como un lenguaje para la representación de modelos de acciones y de problemas de planificación.
- 5. El proceso básico de planificación consiste en un proceso de búsqueda heurística en un espacio de estados.**
- 6. Salida: un plan como una secuencia de acciones que parte de una situación dada van transformando las situaciones o estados y que finalmente producen una situación en la que se alcanza objetivo.**
- 7. Ventaja: se pueden resolver problemas que tienen un fuerte componente combinatorio, por ejemplo, problemas de compañías de logística**
- 8. Vimos que hay dos visiones de la planificación en IA:**
 1. como técnica para resolver problemas de actuación (como el ejemplo justo anterior) y
 2. como técnica para ayudar a la toma de decisiones (detección, predicción,recomendación)
- 9. Limitaciones,**
 1. forma en que exploran el espacio de búsqueda (tamaño exponencial) pueden perder mucho tiempo buscando una solución porque no tienen una guía adecuada.
 2. si pudiéramos inyectar conocimiento humano sobre cómo se resuelven los problemas, podríamos reducir el tiempo de dedicado a la búsqueda.

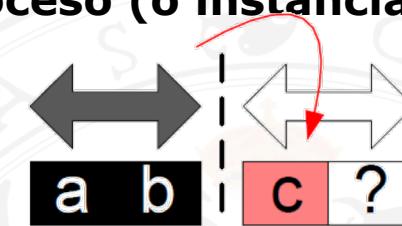
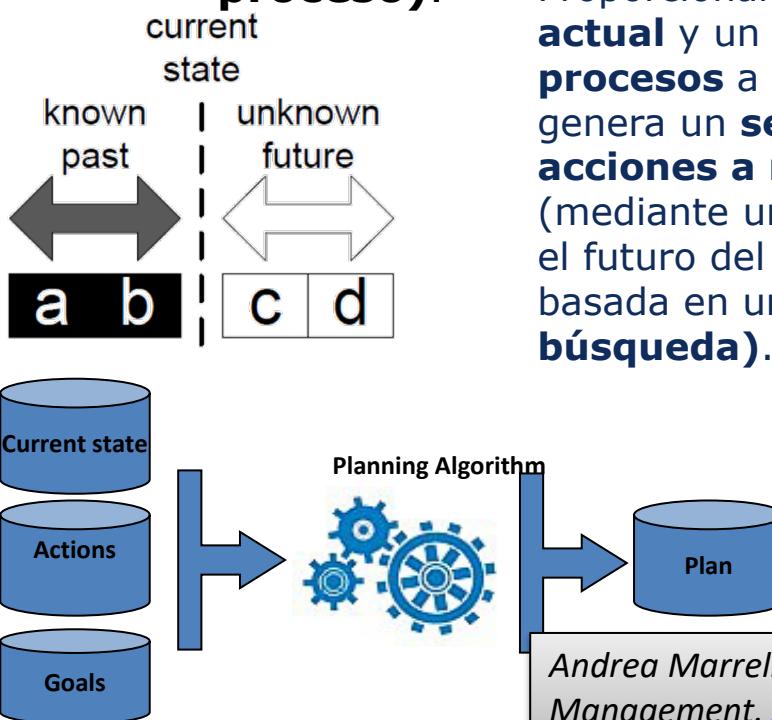
- Visión 2: Planificación automática para generar procesos y ayudar en la toma de decisiones en analítica prescriptiva y minería de procesos.
 - La planificación automática puede verse como la generación de
 - procesos orientados a objetivos (planes)
 - a partir de un modelo de proceso (dominio de planificación).
 - Un **planificador es un generador (o instanciador) de procesos.**
 - Secuencia de actividades, un plan, **es un proceso (o instancia de proceso).**



Analítica
prescriptiva

- Visión 2: Planificación automática para generar procesos y ayudar en la toma de decisiones en analítica prescriptiva y minería de procesos.
 - La planificación automática puede verse como la generación de
 - procesos orientados a objetivos (planes)
 - a partir de un modelo de proceso (dominio de planificación).
 - Un **planificador es un generador (o instanciador) de procesos.**
 - Secuencia de actividades, un plan, **es un proceso (o instancia de proceso).**

Proporcionando el **estado actual** y un **modelo de procesos** a un **planificador** se genera un **secuencia de acciones a recomendar** (mediante una proyección hacia el futuro del estado actual basada en un **proceso de búsqueda**).

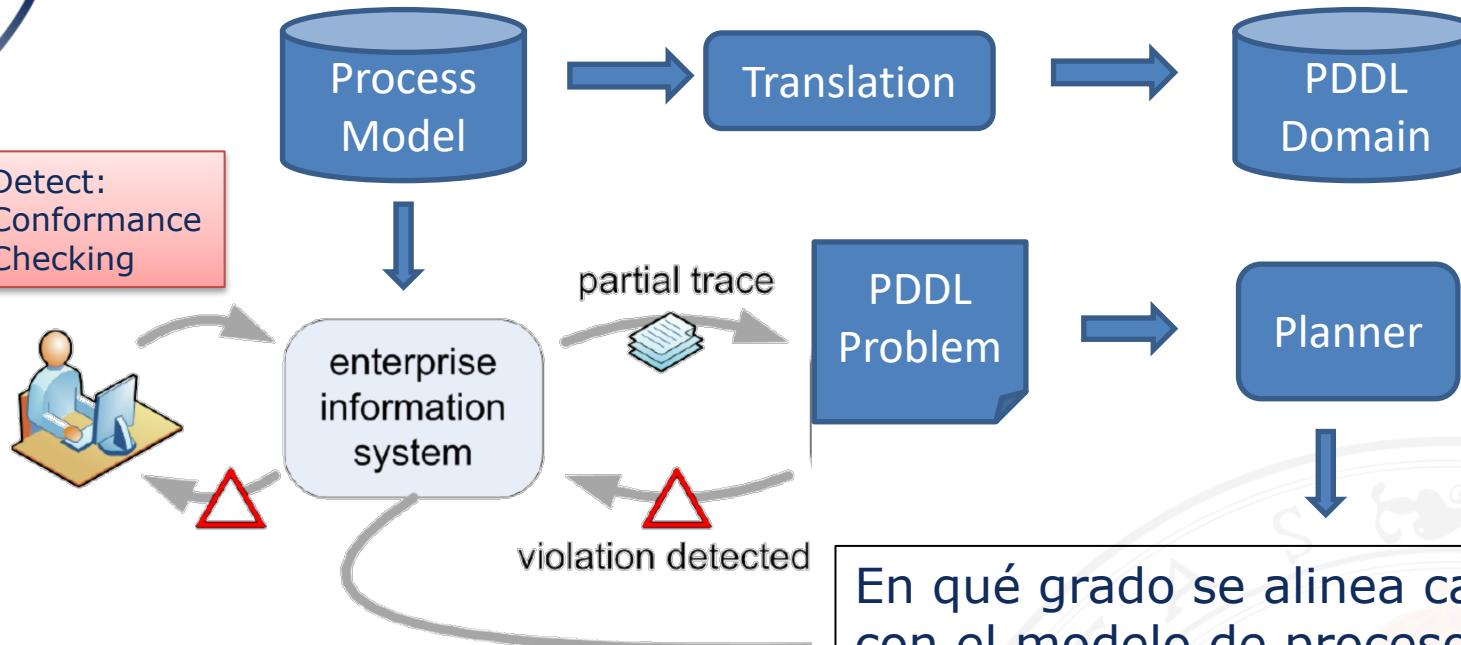


recommend: based on past experiences c is recommended (e.g., to minimize costs)

Analítica prescriptiva

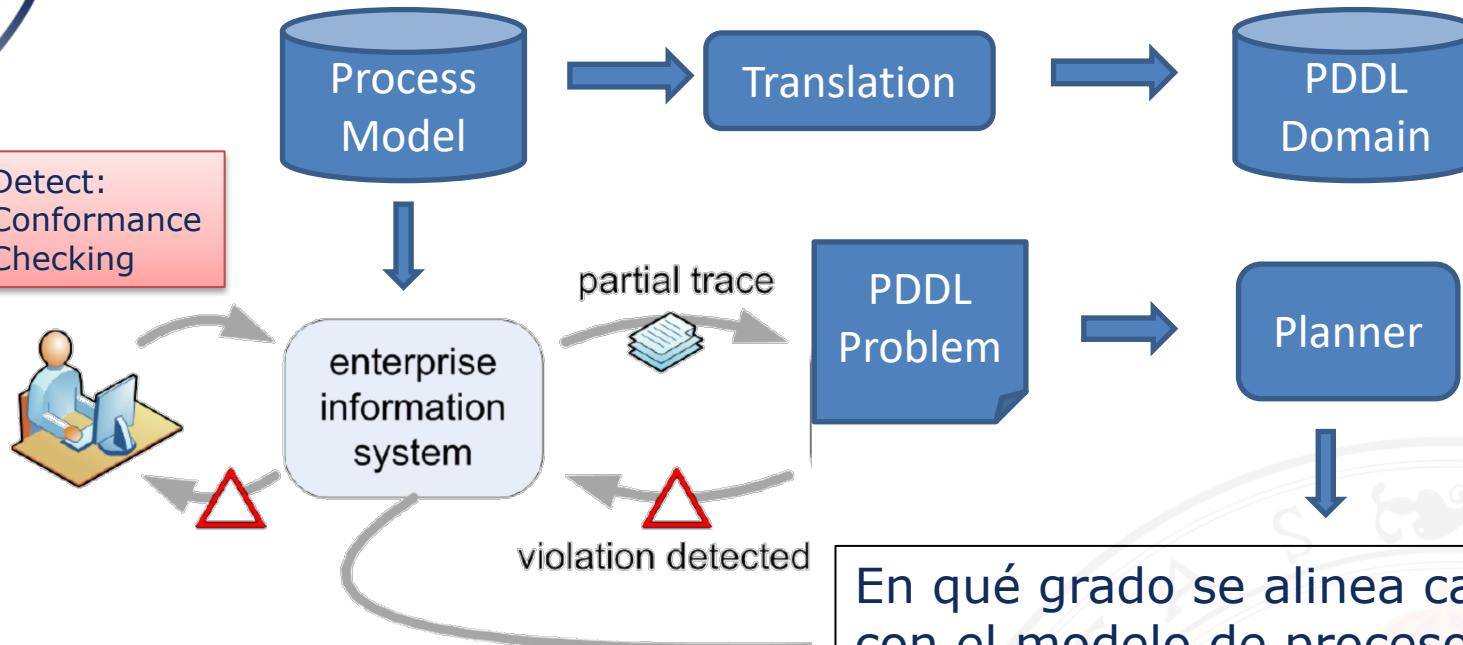
Arturo González-Ferrer . Juan Fernández-Olivares, Luis A. Castillo:From business process models to hierarchical task network planning domains. Knowl. Eng. Rev. 28(2): 175-193 (2013)

Andrea Marrella: Automated Planning for Business Process Management. *J. Data Semant.* 8(2): 79-98 (2019)



En qué grado se alinea cada traza con el modelo de proceso, o , ¿cómo puedo hacer que una traza no alineada si alinee con el modelo de proceso?

[Giuseppe De Giacomo](#), [Fabrizio Maria Maggi](#), [Andrea Marrella](#), [Sebastian Sardiña](#):
Computing Trace Alignment against Declarative Process Models through Planning.
[ICAPS 2016](#): 367-375

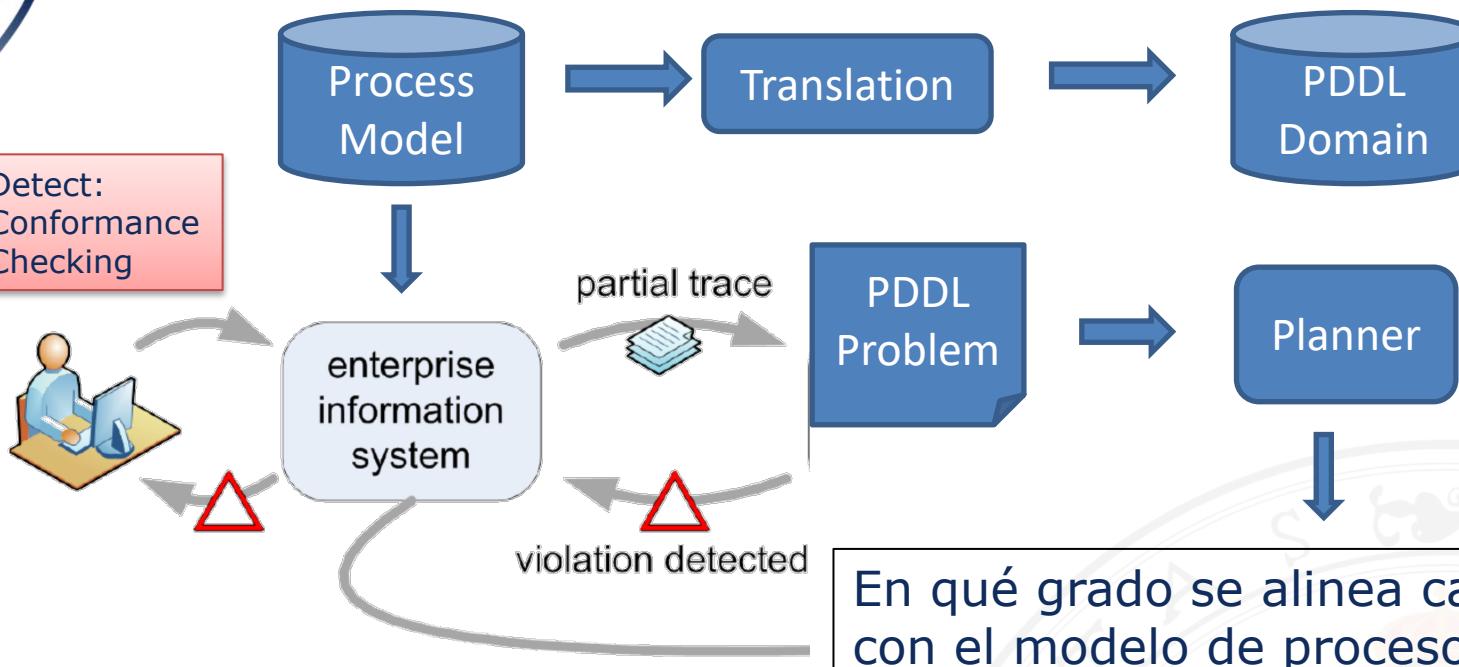


En qué grado se alinea cada traza con el modelo de proceso, o , ¿cómo puedo hacer que una traza no alineada si alinee con el modelo de proceso?

Modelo DECLARE: (Haisjäckl, C., Barba, I., Zugal, S. et al. Understanding Declare models: strategies, pitfalls, empirical results. *Softw Syst Model* 15, 325–352 (2016).
<https://doi.org/10.1007/s10270-014-0435-z>)

- Se define un modelo DECLARE con actividades específicas y restricciones que modelan el comportamiento esperado del proceso. Por ejemplo, puede haber una restricción que especifique que después de una actividad 'A', debe seguir una actividad 'B'.

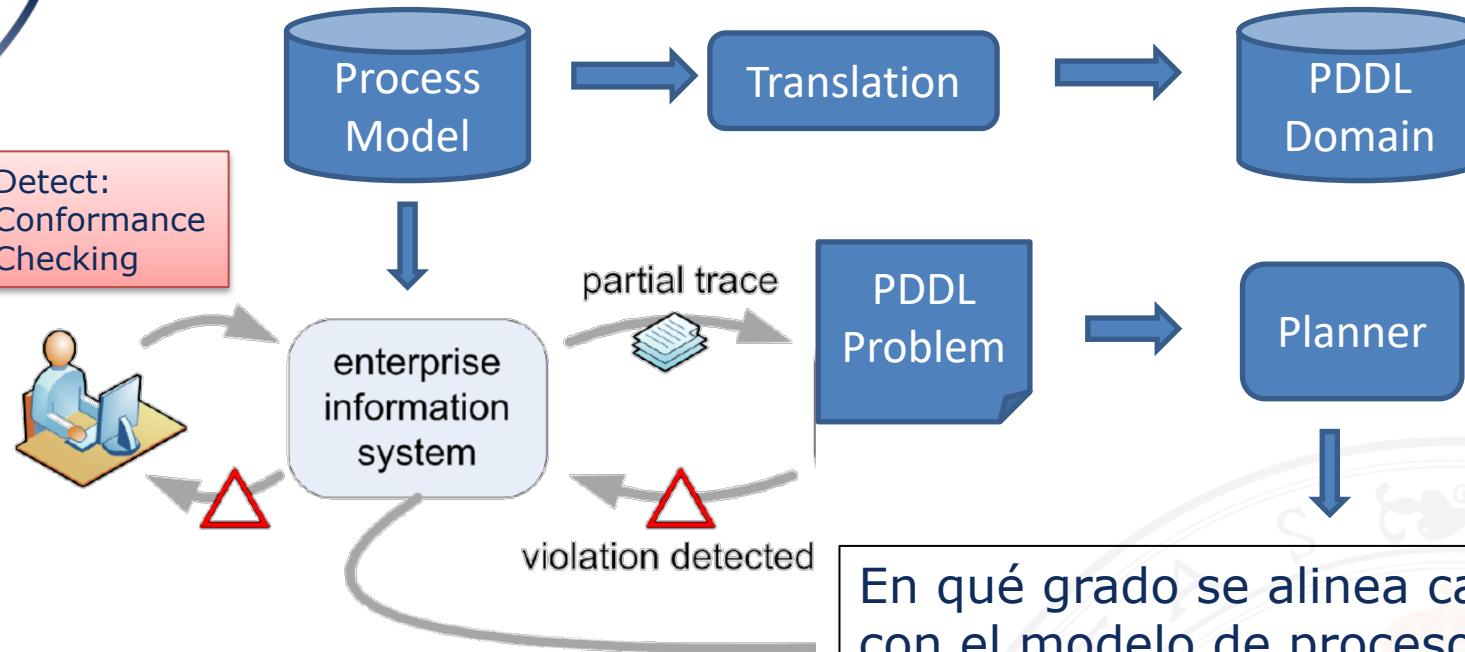
Detect:
Conformance
Checking



En qué grado se alinea cada traza
con el modelo de proceso

Dominio PDDL:

- Tipos:** Se definen tipos de actividades basadas en las actividades identificadas en el proceso.
- Predicados:** Se utilizan para modelar propiedades como la presencia de una actividad en la traza, la posición de la actividad (primera, última), y las conexiones entre actividades (sucesiones).
- Acciones:** Se especifican acciones para agregar o eliminar actividades de la traza. Las acciones reflejan los cambios permitidos que pueden aplicarse a la traza para que cumpla con las restricciones del modelo DECLARE.

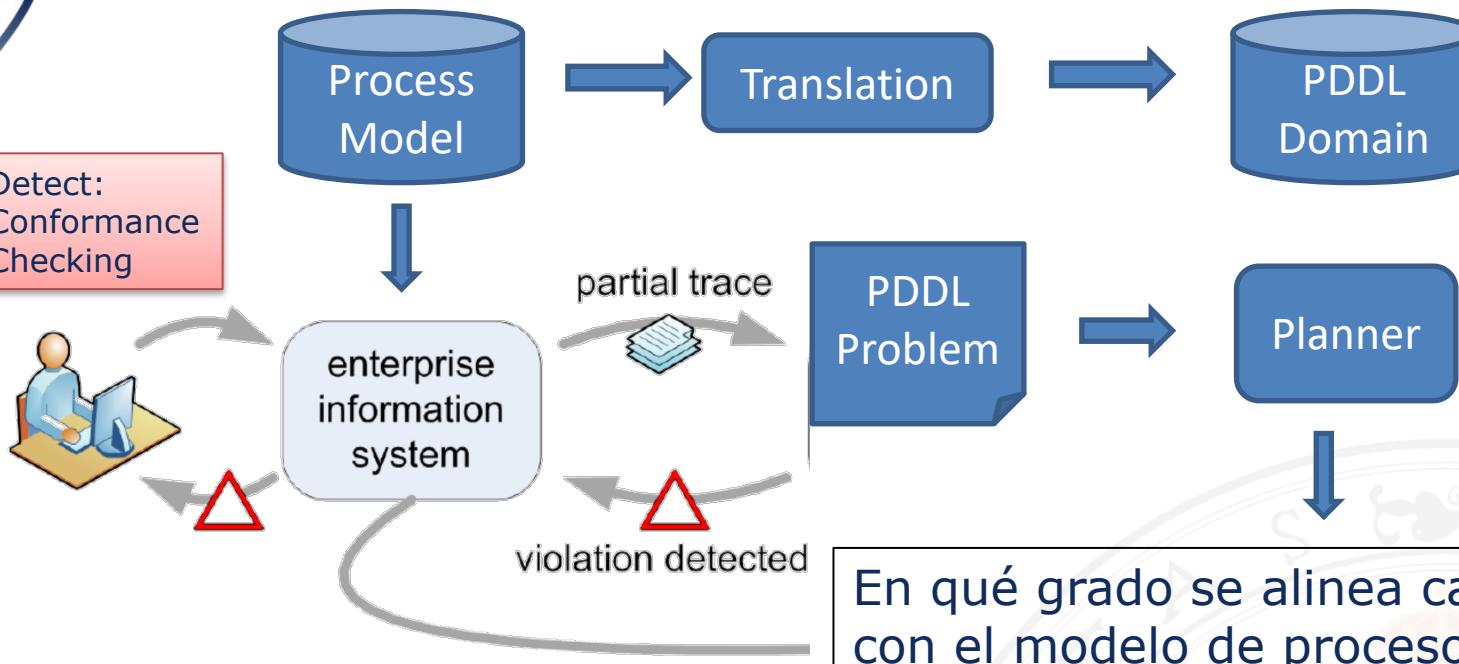


En qué grado se alinea cada traza con el modelo de proceso

Problema PDDL:

- **Estado Inicial:** Describe el estado de la traza antes de cualquier modificación. Por ejemplo, la traza original es $T_0 = \langle a, b, c, a, d \rangle$ y debe cumplir la restricción (descrita en el modelo de proceso) de que 'a' debe ser seguida por 'e', pero **se observa** que 'e' no está presente después de cada 'a'.
- **Objetivo (o meta):** La meta es cumplir con todas las restricciones de DECLARE. Esto se traduce en un conjunto de predicados derivados que representan la satisfacción de cada restricción.
- **Costos:** Se asignan costos a las acciones basadas en la importancia o dificultad para modificar ciertos aspectos de la traza.

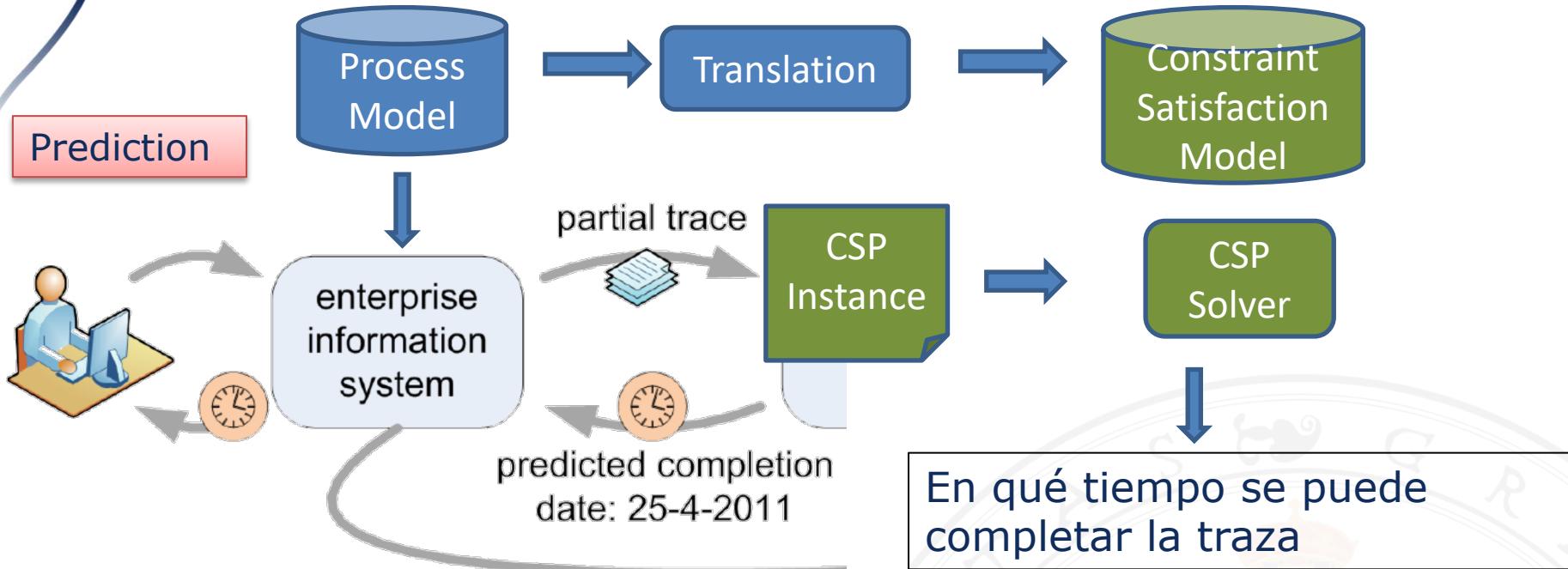
Detect:
Conformance
Checking



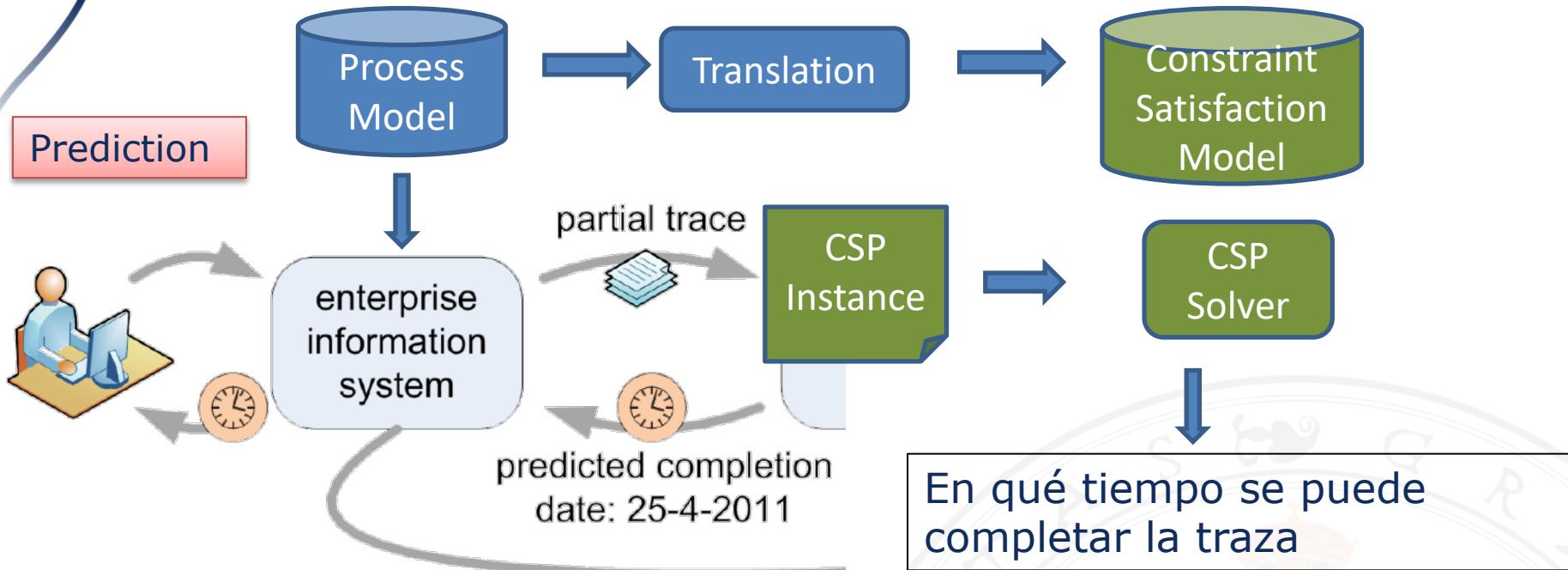
En qué grado se alinea cada traza con el modelo de proceso

Ejecución y Alineación:

- Se ejecuta el planificador con el dominio y el problema definido. El planificador busca secuencias de acciones que **transformen la traza original en una que cumpla con todas las restricciones** del modelo **DECLARE al menor costo posible**.
- Por ejemplo, podría sugerir agregar 'e' después de cada 'a' o modificar la secuencia de actividades para cumplir con otras restricciones como la presencia o ausencia de ciertas actividades en lugares específicos de la traza.



Andres Jiménez Ramirez , Irene Barba , Juan Fernández-Olivares , Carmelo Del Valle , Bárbara Weber *Time prediction on multi-perspective declarative business processes.* [Knowl. Inf. Syst. 57\(3\)](#): 655-684 (2018)



Modelado de Procesos:

- Se utiliza un lenguaje basado en restricciones (usados para modelar CSPs: Constraint Satisfaction Problems) para modelar los procesos de negocio. Este modelo considera diferentes aspectos como las actividades, el tiempo, los recursos, los datos y el flujo de control.

Ejemplo Ilustrativo: Salón de Belleza

Contexto: Imagina un día típico en un salón de belleza donde se realizan diferentes servicios como cortes de cabello, coloración y manicura. Cada servicio tiene un tiempo estimado de realización y requiere asignación de recursos específicos (por ejemplo, estilistas y equipos de corte de pelo).

Traza de Proceso: Supongamos que hasta el momento se han realizado las siguientes actividades:

- **Actividad A (Recepción):** El cliente ha llegado y ha sido registrado.
- **Actividad B (Corte de cabello):** El cliente ha recibido un corte de cabello.

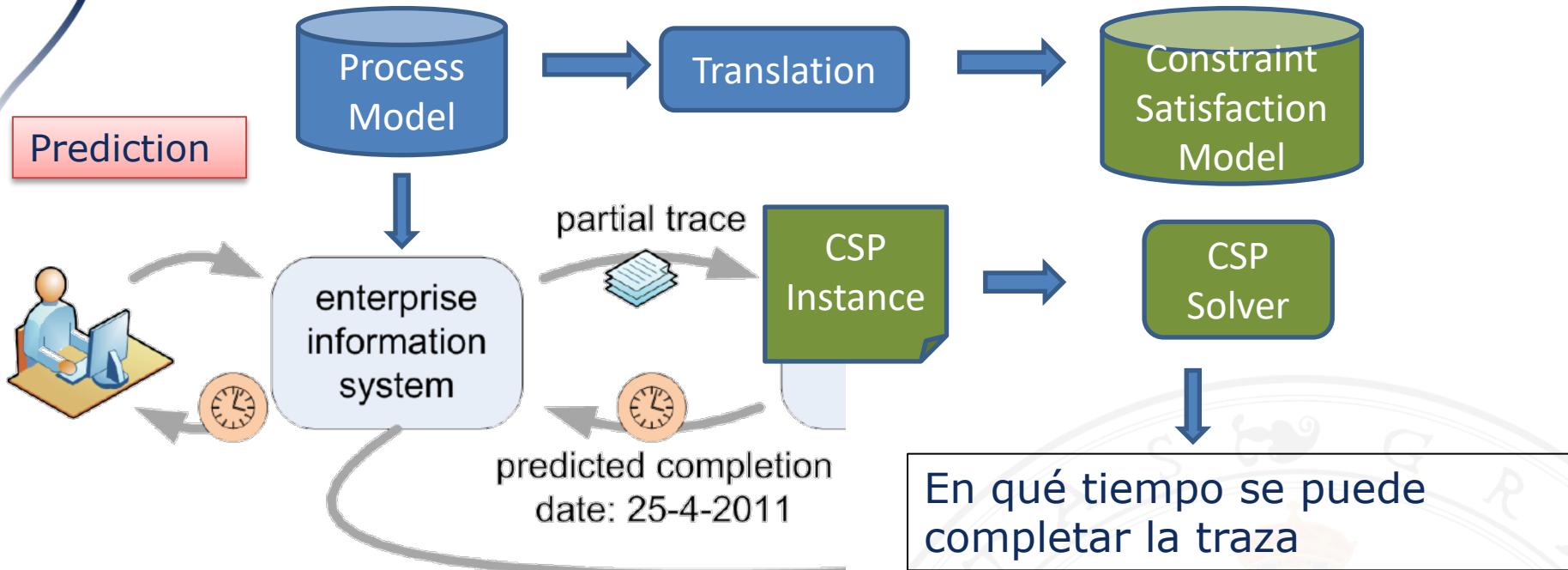
Representación como CSP:

1. Definición de Variables para A, B: actividades

1. **A_start, A_end:** Tiempo de inicio y fin para la Actividad A.
2. **B_start, B_end:** Tiempo de inicio y fin para la Actividad B.
3. **Resource_A, Resource_B:** Recursos asignados a las actividades A y B respectivamente.

2. Establecimiento de Restricciones:

1. **A_end ≤ B_start:** La Actividad B (corte de cabello) no puede comenzar hasta que la Actividad A (recepción) haya terminado.
2. **Resource constraints:** Si la actividad A y B comparten recursos, como un estilista específico, se debe garantizar que este recurso no sea doblemente asignado en el mismo intervalo de tiempo.

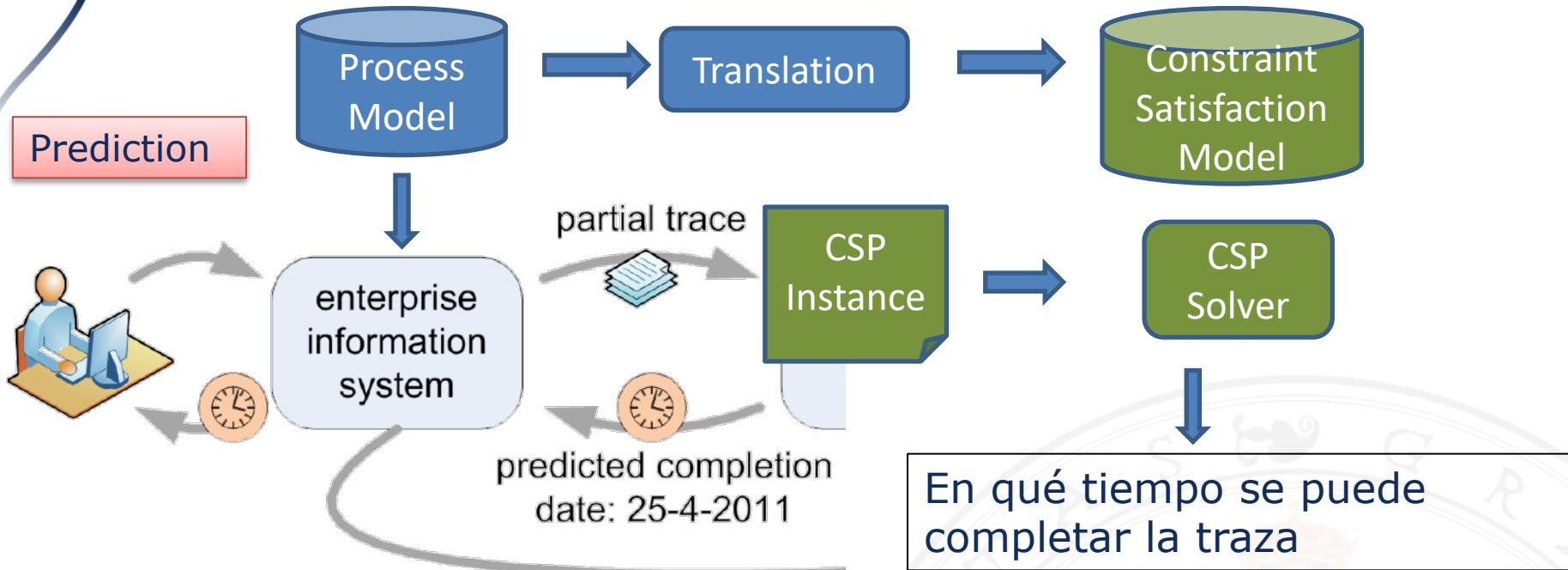


1. Definición de Variables:

- Cada actividad en la traza se asocia con un conjunto de variables CSP. Estas variables pueden incluir el tiempo de inicio (**start_time**), el tiempo de finalización (**end_time**), y el recurso asignado (**resource**). Además, se pueden incluir variables que indican si una actividad específica debe ejecutarse (**execute**).

2. Establecimiento de Restricciones:

- Las restricciones se definen en función de las dependencias entre actividades, los requisitos de recursos y las limitaciones temporales específicas del modelo de proceso. Por ejemplo, si una actividad 'B' no puede comenzar hasta que la actividad 'A' haya terminado, se crea una restricción que relaciona los tiempos de finalización de 'A' y los tiempos de inicio de 'B'.
- Otras restricciones pueden involucrar la asignación de recursos, asegurando que no se asignen más actividades a un recurso de las que puede manejar simultáneamente.

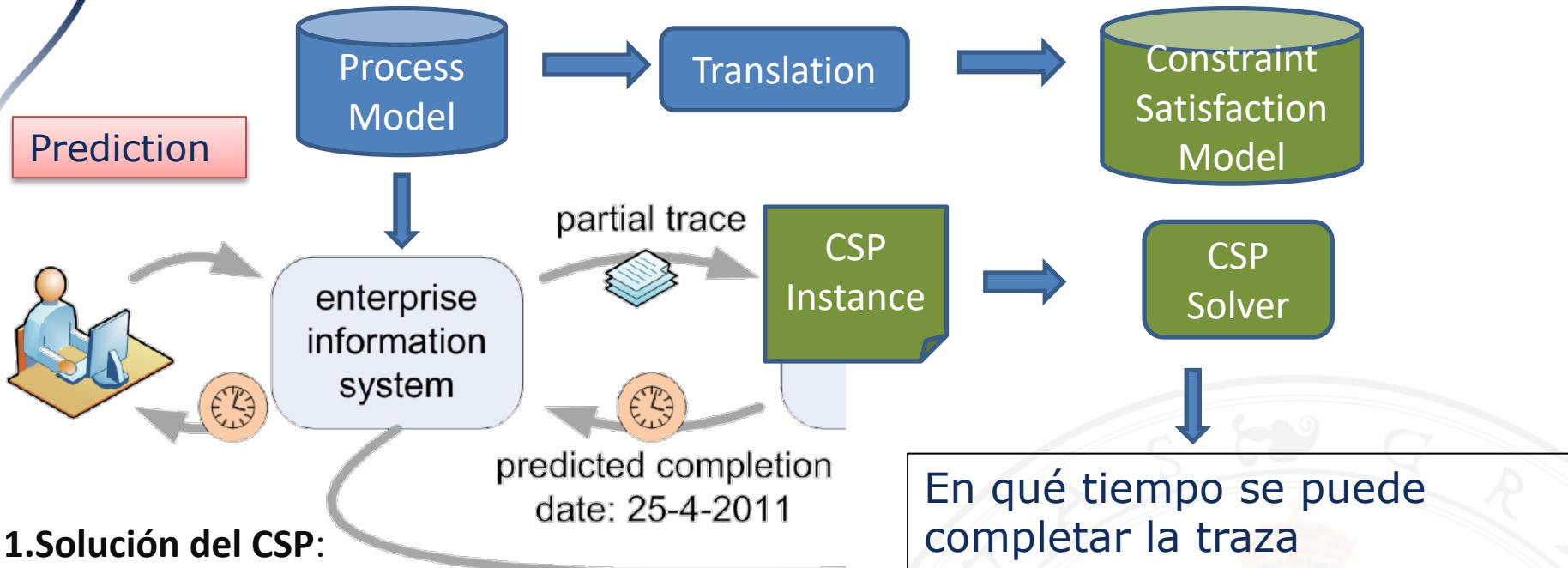


Generación del Plan de Ejecución:

- Basado en el modelo, el sistema genera un plan de ejecución optimizado para un día específico, considerando las citas programadas y la disponibilidad de empleados. Este plan indica la secuencia óptima de actividades y la asignación de recursos.

Optimización:

- El objetivo del CSP es encontrar valores para las variables que satisfagan todas las restricciones. Además, se puede definir una función objetivo, como minimizar el tiempo total o maximizar la utilización de recursos, para guiar la búsqueda hacia soluciones óptimas.



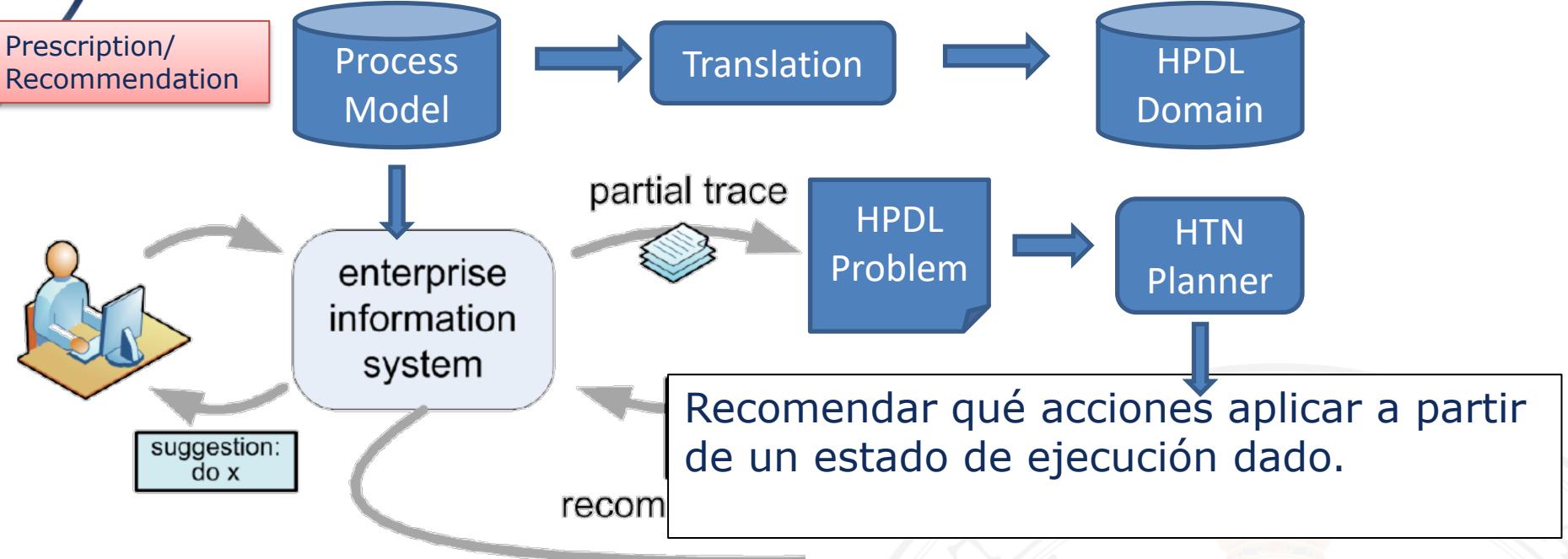
1. Se aplican algoritmos de resolución de CSP para encontrar los valores óptimos de **A_start**, **A_end**, **B_start**, **B_end**, **Resource_A**, y **Resource_B** que satisfagan todas las restricciones y optimicen la función objetivo.

Función Objetivo: Minimizar B_end: El objetivo podría ser minimizar el tiempo total del servicio para mejorar la satisfacción del cliente, asegurando que el cliente complete su visita al salón lo más rápido posible sin comprometer la calidad del servicio.

Ejecución de la Solución

Suponiendo que se encontró la siguiente solución óptima:

- **A_start = 09:00, A_end = 09:15** (El cliente llega y es registrado en 15 minutos).
- **B_start = 09:15, B_end = 09:45** (El corte de cabello toma 30 minutos).
- **Resource_A = Recepcionista, Resource_B = Estilista 1.**

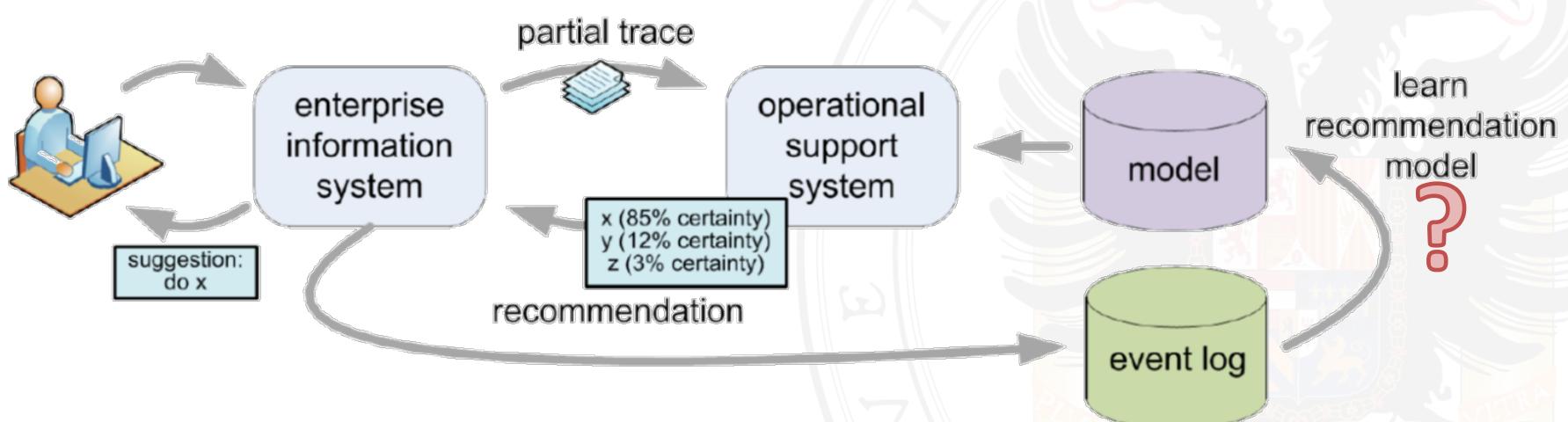
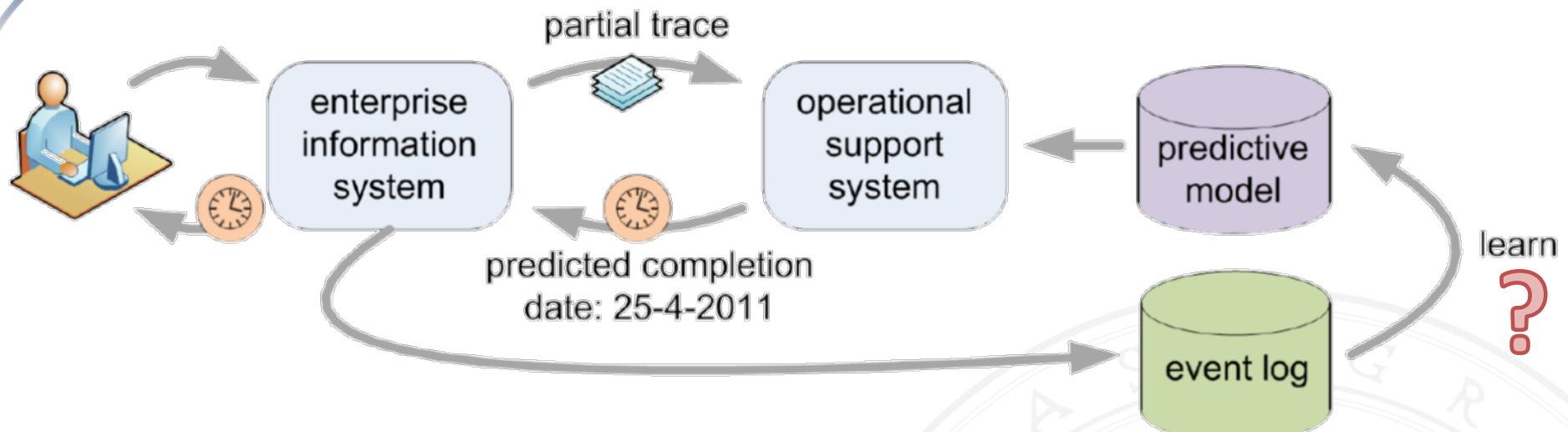


Modelos de proceso representados en BPMN

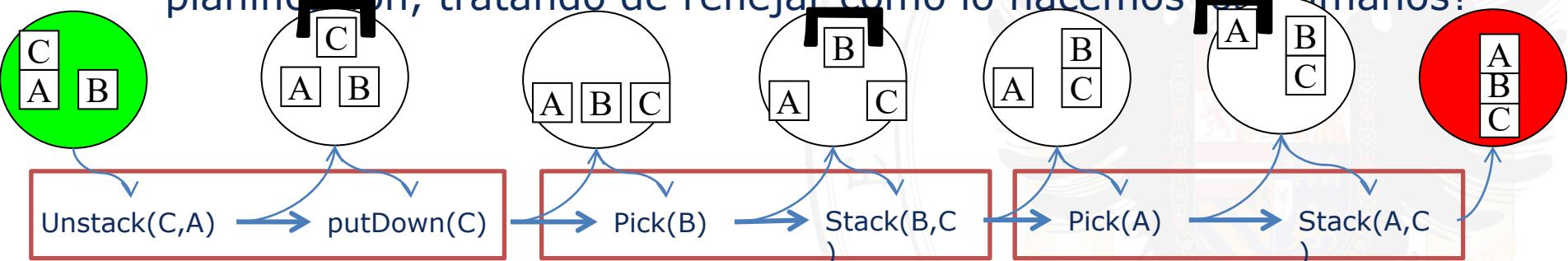
[Arturo González-Ferrer](#) . [Juan Fernández-Olivares](#), [Luis Castillo](#): *From business process models to hierarchical task network planning domains.* [Knowl. Eng. Rev.](#) 28(2): 175-193 (2013)

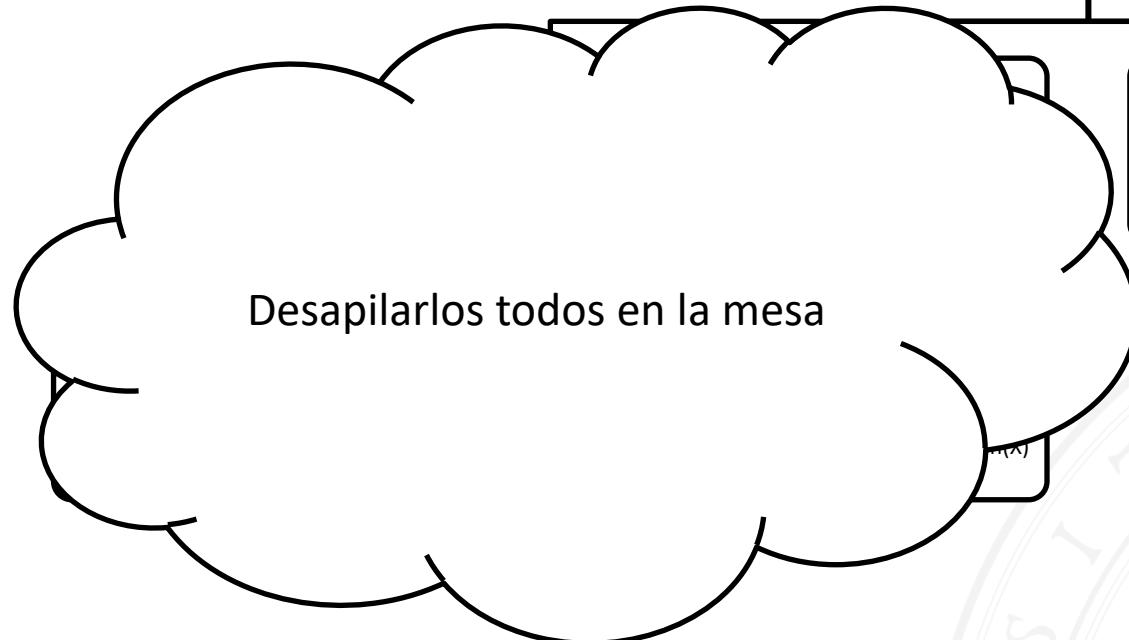
Modelos de proceso representados en Asbru (Protocolos clínicos en Medicina)

[Arturo González-Ferrer](#) . [Annette ten Teije](#) . [Juan Fernández-Olivares](#), [Krystyna Milian](#): *Automated generation of patient-tailored electronic care pathways by translating computer-interpretable guidelines into hierarchical task networks.* [Artif. Intell. Medicine](#) 57(2): 91-109 (2013)



- Ventajas:
 - Resolución de problemas combinatorios.
 - Puede representar y solucionar algunos problemas para predicción/detección en Minería de Procesos
- Limitaciones:
 - Pueden explorar caminos claramente desechar.
 - Centrado en analizar relaciones causa-efecto, dificultad en representar otras relaciones entre acciones, como patrones de proceso comunes, habituales en actividades humanas.
 - Dificultades para representar **modelos normativos**
- ¿Podemos mejorar la forma en que se resuelven problemas de planificación, tratando de reflejar cómo lo hacemos los humanos?

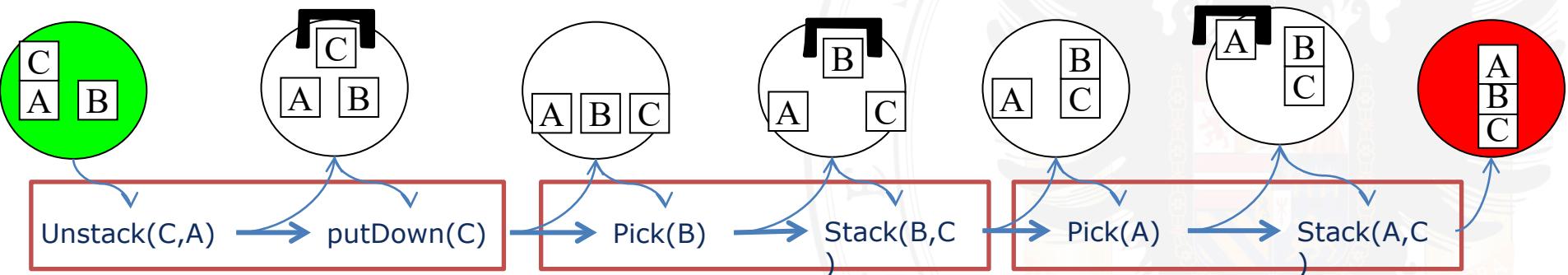


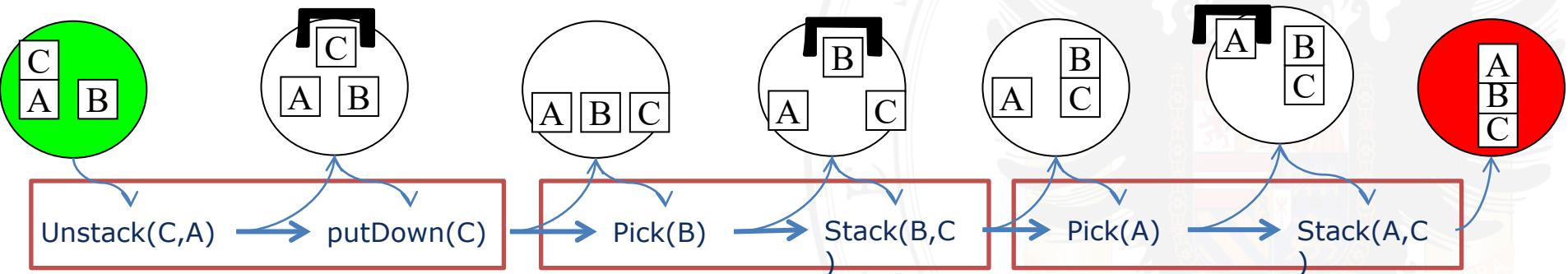
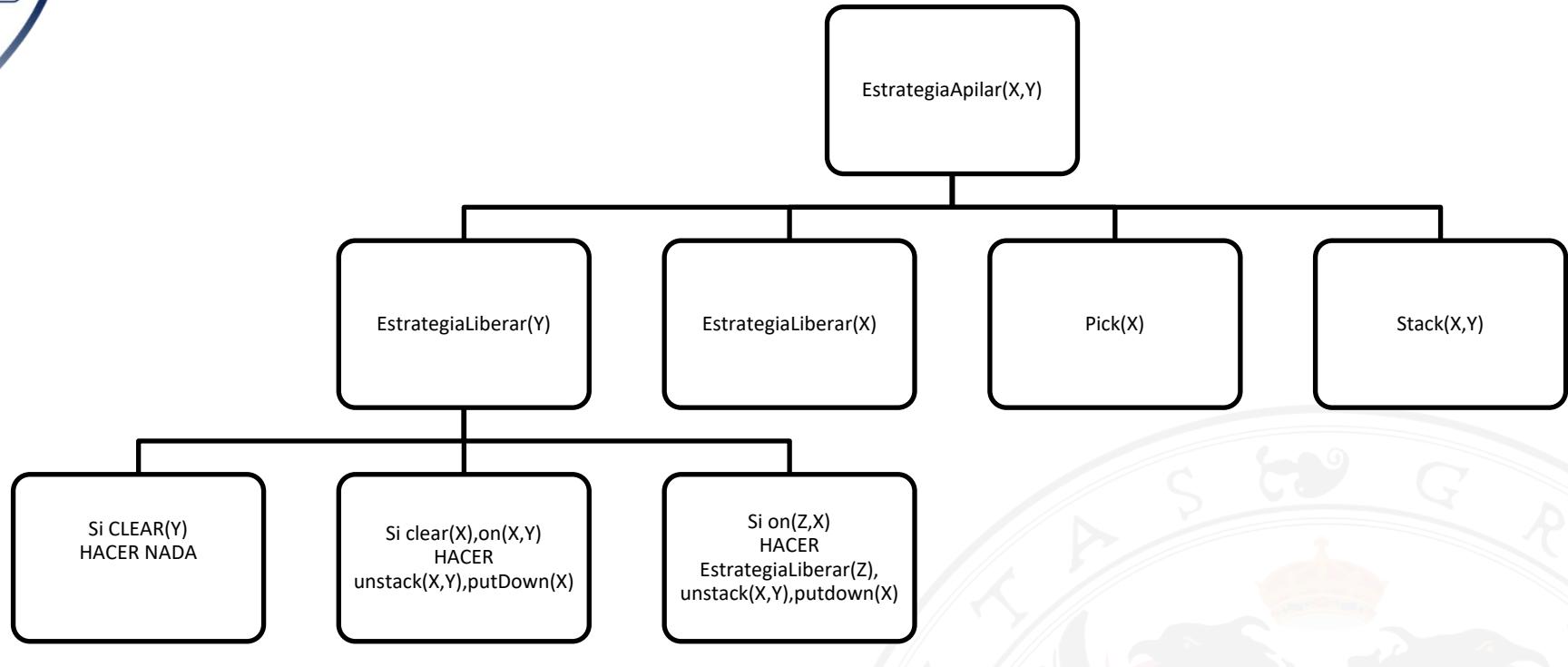


EstrategiaApilar(X,Y)

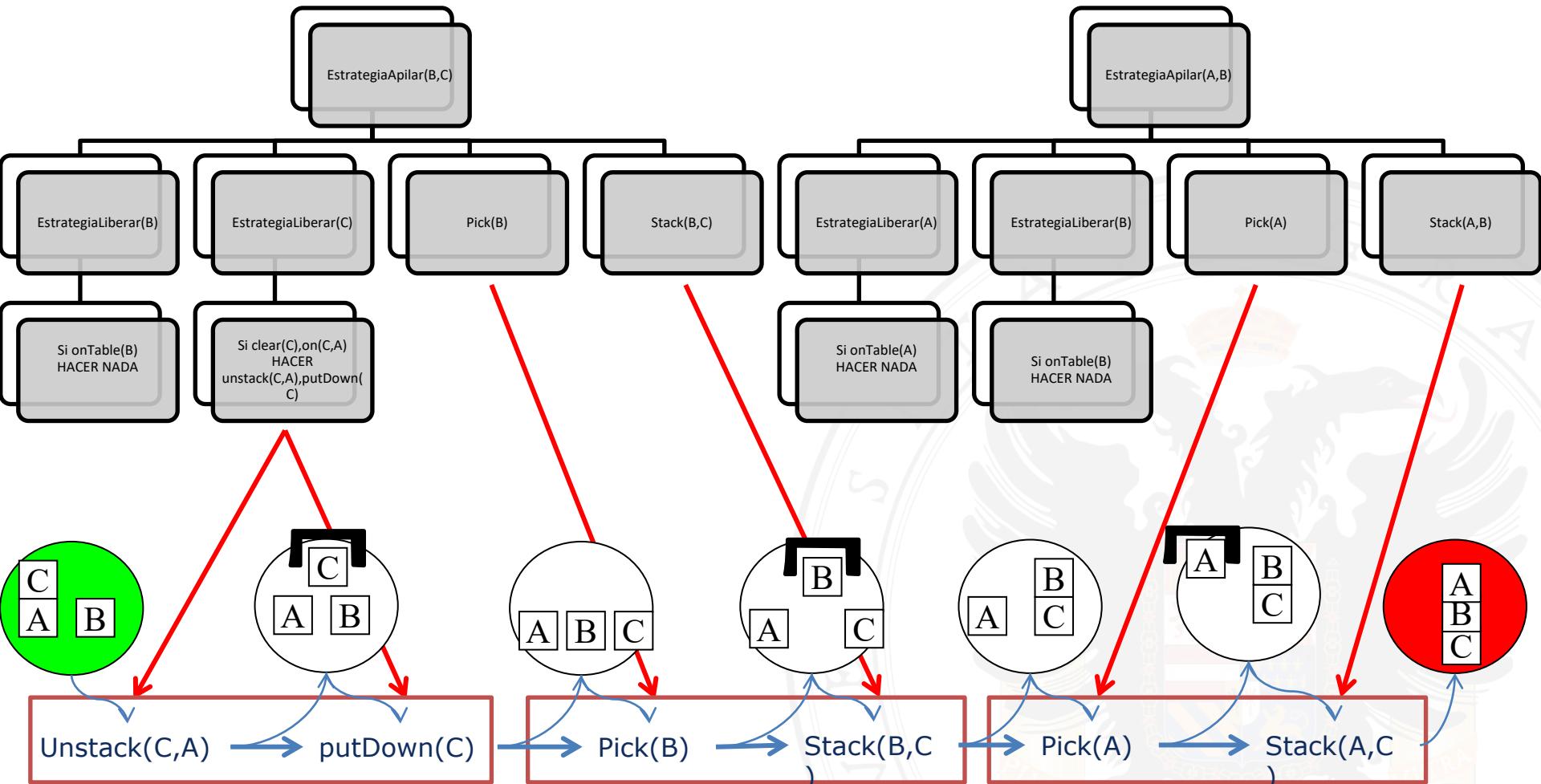
Pick(X)

Stack(X,Y)

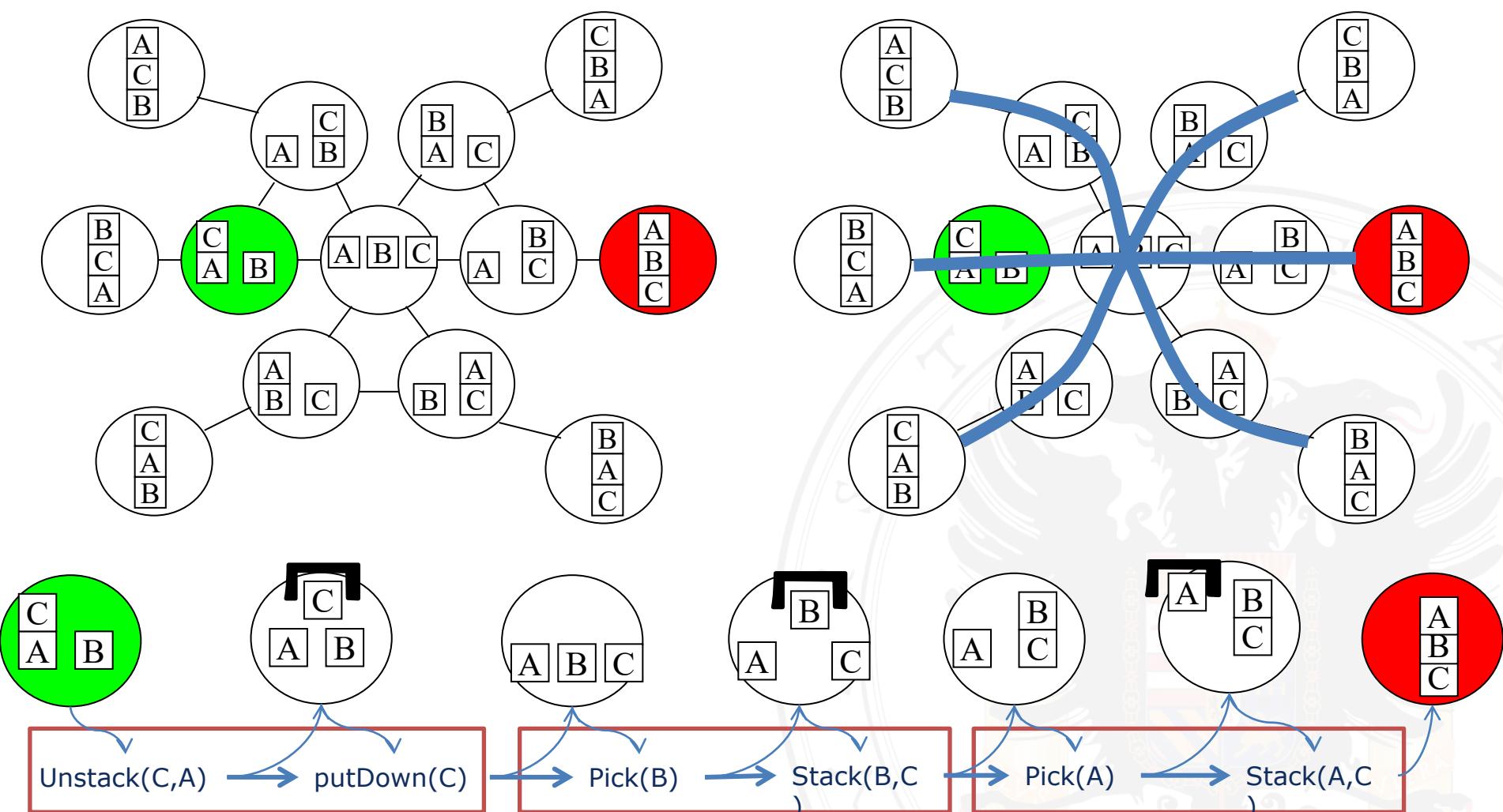




Planificación jerárquica: técnica de planificación que permite representar estrategias a distintos niveles de abstracción para resolver problemas de planificación.

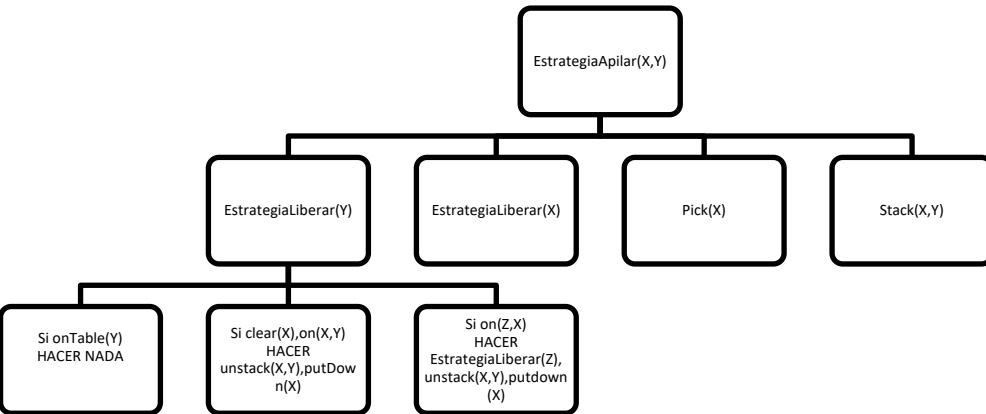


- Ventaja 1: Podemos reducir el espacio de búsqueda y conseguir procesos de resolución más eficientes y guiados.



- Ventaja2:

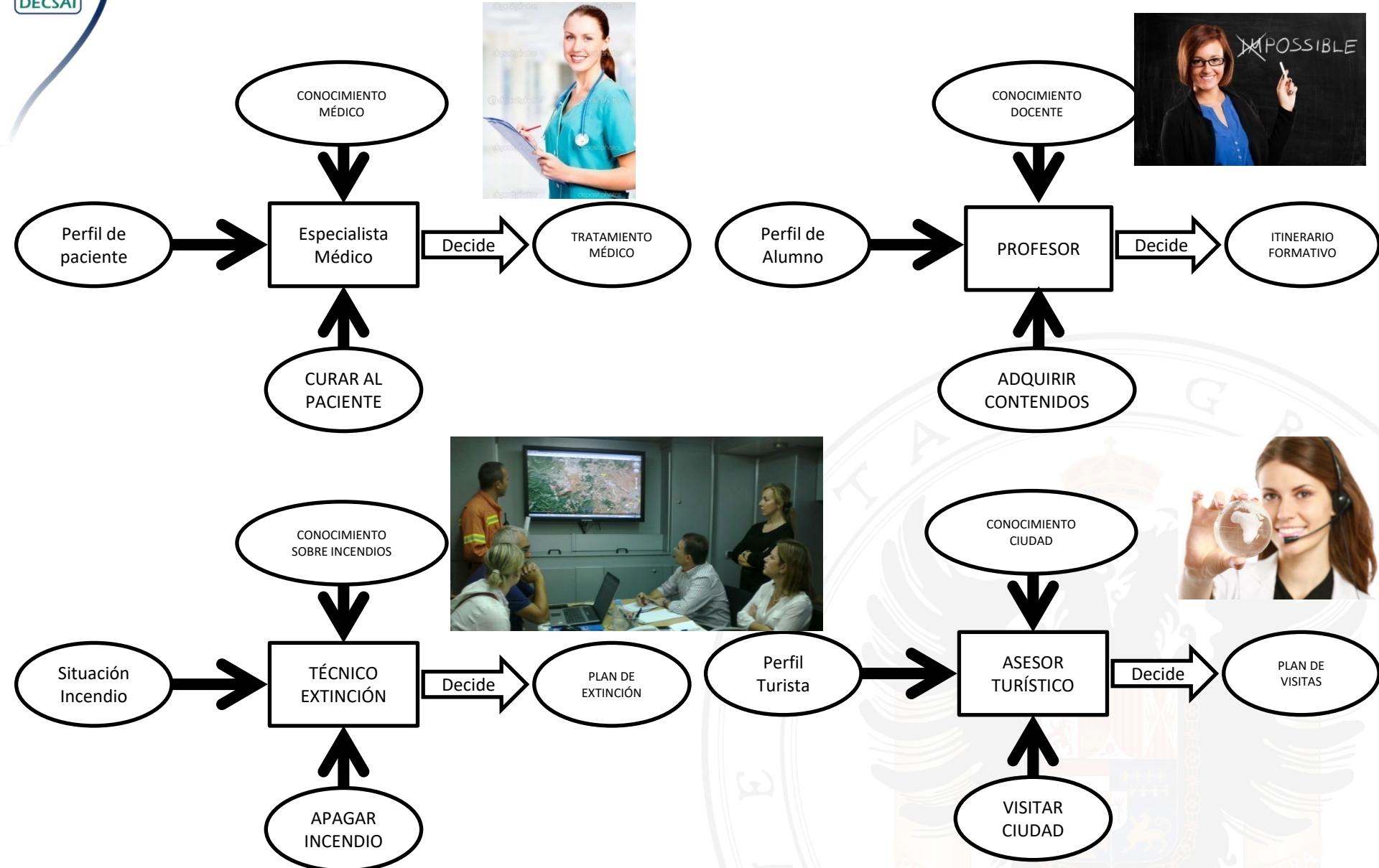
- Podemos representar conocimiento experto humano para describir estrategias para resolver problemas.
- Podemos generar planes (procesos) adaptados a la forma en que los humanos resuelven problemas



- Ventaja 3: podemos resolver más problemas y más próximos a la realidad.



Planificación jerárquica: CASOS DE USO





Juan Fdez-Olivares, [Eva Onaindia](#), [Luis A. Castillo](#), [Jaume Jordán](#), [Juan A. Cázar](#): Personalized conciliation of clinical guidelines for comorbid patients through multi-agent planning. [Artif. Intell. Medicine](#) 96: 167-186 (2019)

[Inmaculada Sánchez-Garzón](#), [Arturo González-Ferrer](#) . Juan Fernández-Olivares: A knowledge-based architecture for the management of patient-focused care pathways. [Appl. Intell.](#) 40(3): 497-524 (2014)



[Luis A. Castillo](#), [Lluvia Morales](#), [Arturo González-Ferrer](#) . Juan Fernández-Olivares, [Daniel Borrajo](#) . [Eva Onaindia](#) : Automatic generation of temporal planning domains for e-learning problems. [J. Sched.](#) 13(4): 347-362 (2010)

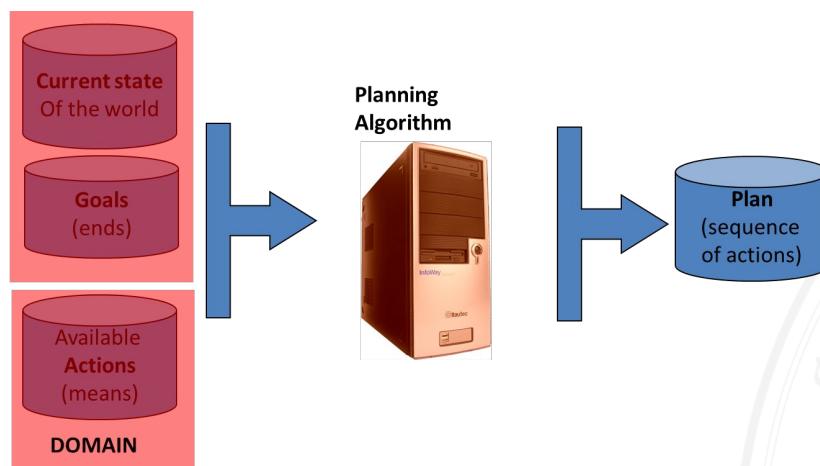


Juan Fernández-Olivares, [Luis A. Castillo](#), [Óscar García-Pérez](#), [Francisco Palao](#): Bringing Users and Planning Technology Together. Experiences in SIADEX. [ICAPS 2006](#): 11-20



Luis A. Castillo, [Eva Armengol](#), [Eva Onaindia](#) . [Laura Sebastia](#) . [Jesús González-Boticario](#) . [Antonio Rodríguez](#), [Susana Fernández](#) . [Juan D. Arias](#), [Daniel Borrajo](#) : samap: An user-oriented adaptive system for planning tourist visits. [Expert Syst. Appl.](#) 34(2): 1318-1332 (2008)

- Planificación HTN (Hierarchical Task Networks): una técnica de planificación jerárquica en la que el dominio se representa a partir de:
 - Un conjunto de **tareas compuestas**
 - Para las que se especifican alternativas de descomposición mediante **métodos de descomposición**
 - Un conjunto de **tareas primitivas**.

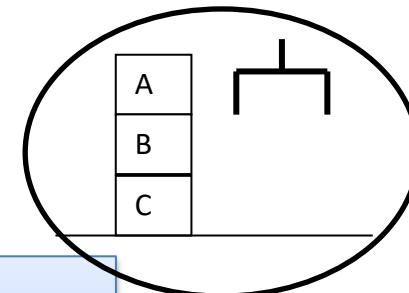


Se mantienen los mismos elementos que en planificación clásica (**dominio, estado actual, objetivo, algoritmo de planificación y plan**) pero se **representan** de distinta manera.

Georgievski, Ilche, and Marco Aiello. "HTN Planning: Overview, Comparison, and beyond." *Artificial Intelligence* 222 (May 2015): 124–56.
doi:10.1016/j.artint.2015.02.002.

- Acción primitiva

Representa una acción del nivel de abstracción inferior, cuya ejecución produce un cambio en el estado



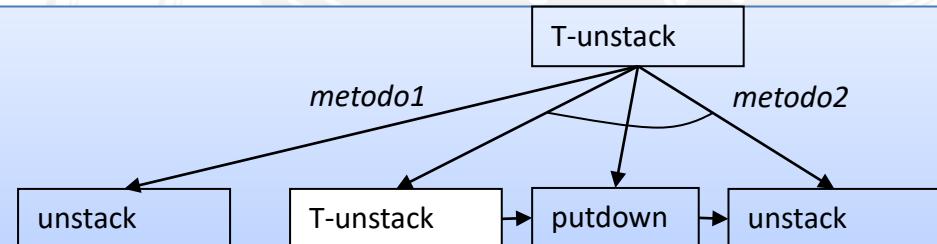
```
(:action unstack
  :parameters (?x ?y - block)
  :condition (and (handempty) (clear ?x))
  :effect (and (not (handempty)) (holding ?x) (clear ?y)
                (not (clear ?x))))
```

- Tarea compuesta y método de reducción (o descomposición).

Tarea de alto nivel que debe llevarse a cabo con la intervención de varias tareas de nivel inferior

Método: modo de llevar a cabo una tarea, representado como un conjunto de subtareas y relaciones de orden entre ellas. Un método es aplicable cuando sus precondiciones son ciertas en el estado.

```
(:task T-unstack
  :parameters (?x ?y - block)
  (:method one
    :precondition (clear ?x)
    :tasks (unstack ?x ?y))
  (:method two
    :precondition (on ?z ?x)
    :tasks ((T-unstack ?z ?x) (putdown ?z) (unstack ?x ?y))))
```



```
(:task Tstack
:parameters (?x ?y - block)
(:method clear
:precondition (clear ?y)
:tasks ((Tpickup ?x) (stack ?x ?y)))
(:method not_clear
:precondition (on ?z ?y)
:tasks ((Tunstack ?z ?y)
(Tputdown ?z) (Tpickup ?x) (stack ?x ?y)))))

(:task Tunstack
:parameters (?x ?y - block)
(:method clear
:precondition (clear ?x)
:tasks (unstack ?x ?y))
(:method not_clear
:precondition (on ?z ?x)
:tasks ((Tunstack ?z ?x) (Tputdown ?z) (unstack ?x
?y))))
```

```
(:action pickup
:parameters (?x - block)
:precondition (and (ontable ?x)(clear ?x)(handempty))
:effect (and (not (ontable ?x)) (not (clear ?x))(not (handempty)) (holding ?x)))

(:action putdown
:parameters (?x - block)
:precondition (holding ?x)
:effect (and (ontable ?x) (clear ?x) (handempty) (not (holding ?x))))
```

```
(:task Tpickup
:parameters (?x - block)
(:method clear
:precondition (and (clear ?x) (ontable ?x))
:tasks (pickup ?x))
(:method clear2
:precondition (and (clear ?x) (on ?x ?z))
:tasks (unstack ?x ?z))
(:method not_clear
:precondition (on ?z ?x)
:tasks ((Tunstack ?z ?x) (Tputdown ?z) (pickup ?x)))))

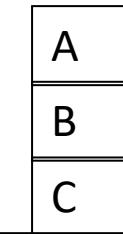
(:task Tputdown
:parameters (?x - block)
(:method clear
:precondition (holding ?x)
:tasks (putdown ?x))
(:method ocupado
:precondition ()
:tasks ((Tpickup ?x) (putdown ?x))))
```

```
(:action stack
:parameters (?x ?y - block)
:precondition (and (holding ?x)(clear ?y))
:effect (and (not (holding ?x)) (not (clear ?y)) (clear ?x)(on ?x ?y) (handempty)))

(:action unstack
:parameters (?x ?y - block)
:precondition (and (handempty) (clear ?x)(on ?x ?y))
:effect (and (holding ?x)(clear ?y)(not (clear ?x)) (not (on ?x ?y)) (not (handempty)))
```

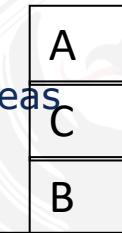
- Estado inicial

(on A B) (on B C) (ontable C)
(clear A) (handempty)



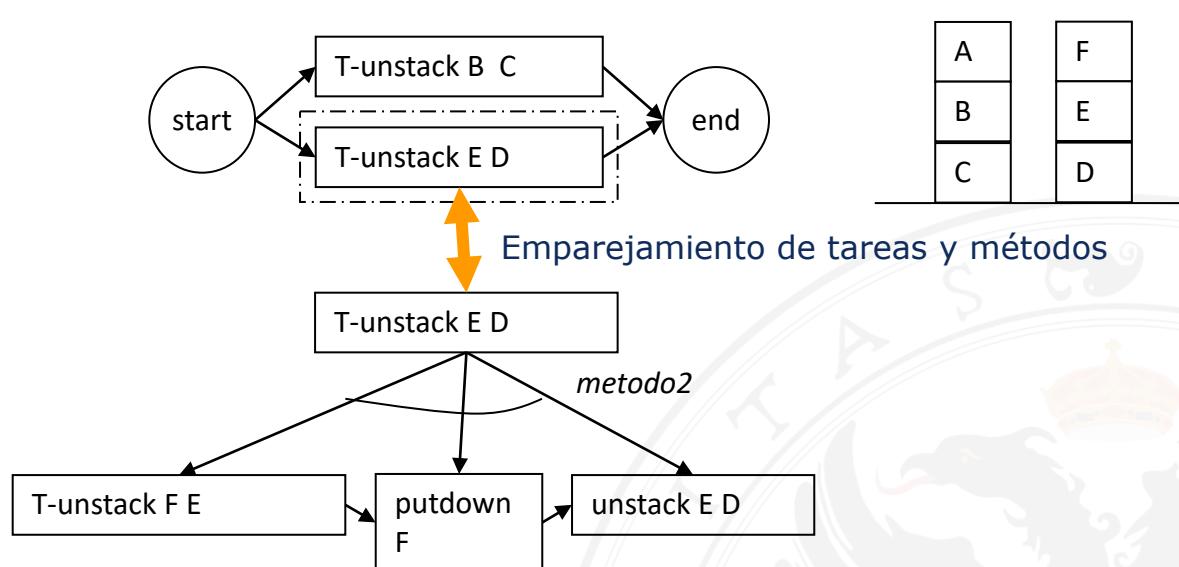
- Objetivo

- Clásico (No HTN): (on C B)(on A C)
 - Interpretación: se desea **alcanzar** dos objetivos como dos estados
 - **No se pueden** especificar estrategias
- HTN: (Apilar C B) (Apilar A C)
 - Interpretación: se desea **desempeñar** dos tareas, cada una destinada a alcanzar un objetivo.
 - Se especifica **una estrategia** para alcanzar uno o varios objetivos

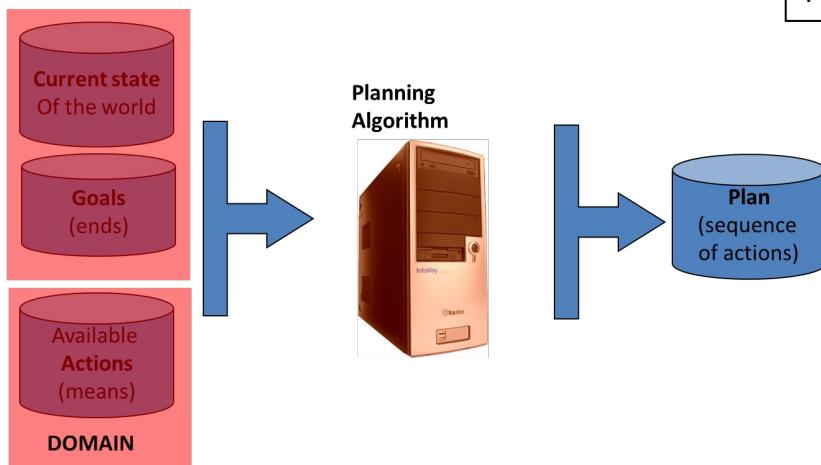


Objetivo :

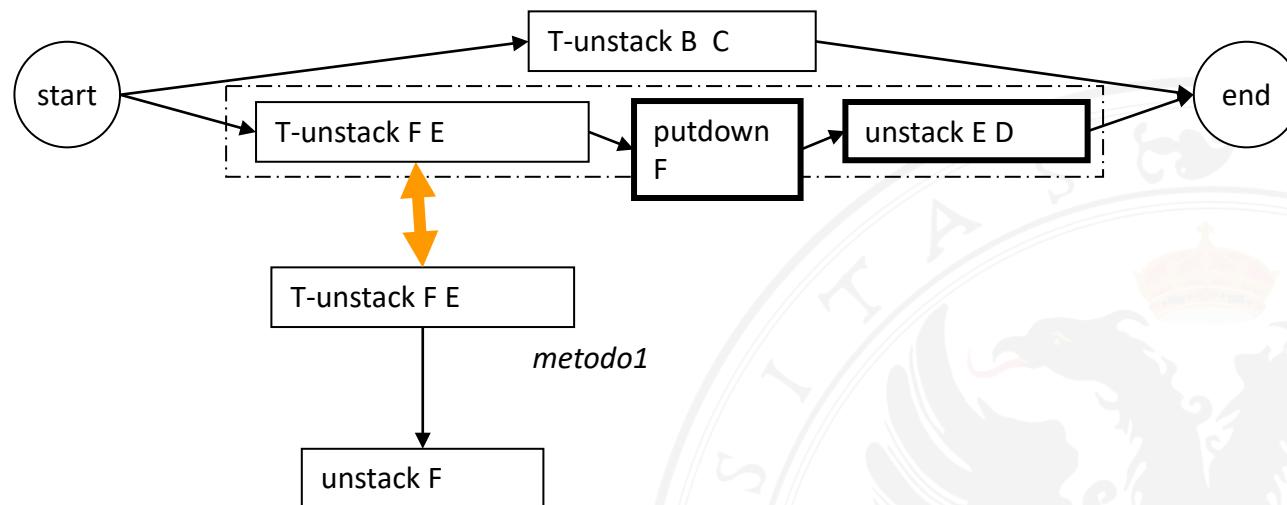
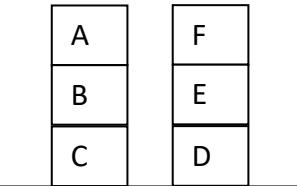
- Red de tareas: conjunto de tareas de alto nivel ordenadas.



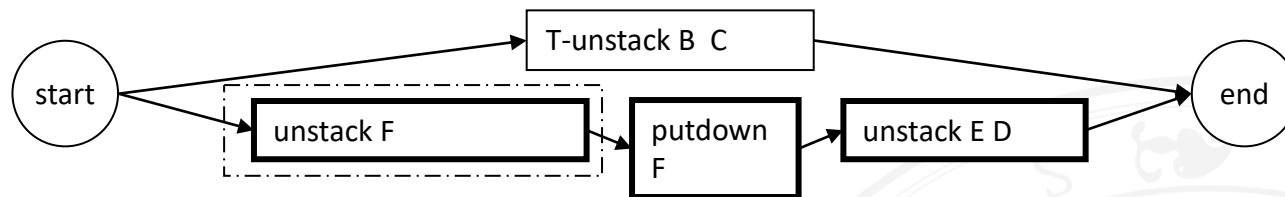
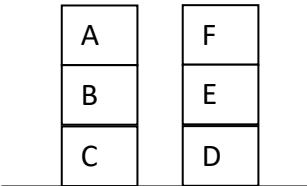
- Proceso de planificación: reducir (descomponer) tareas compuestas aplicando métodos de descomposición.



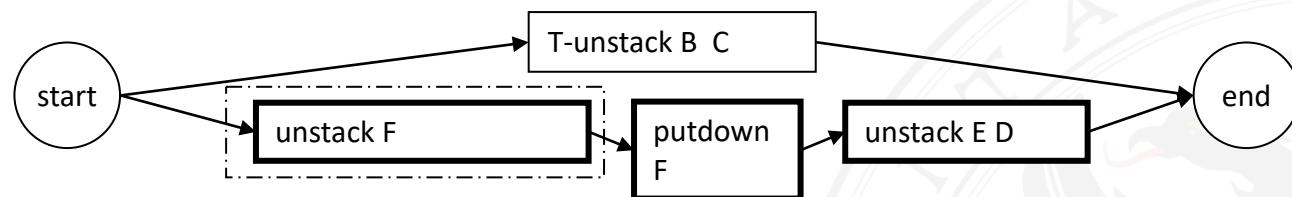
- Red de tareas reducida:
 - La reducción continua de forma recursiva mientras haya tareas compuestas en una red de tareas.



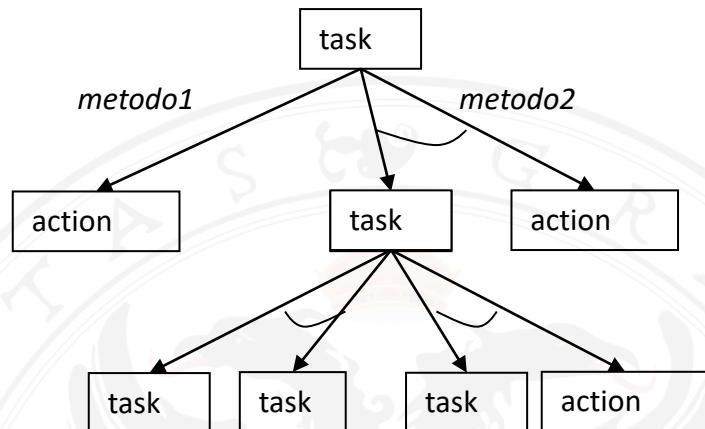
- Planning goal
 - El proceso de reducción de una tarea finaliza cuando todas las subtareas son primitivas.



- Un plan es válido cuando:
 - Todos los nodos de la red de tareas resultante de las reducciones se han convertido en acciones primitivas.
 - Se han obtenido con descomposiciones válidas:
 - Las precondiciones de métodos se han respetado
 - Las precondiciones de las tareas primitivas se han respetado
 - Las variables se han instanciado correctamente
 - Las restricciones de orden entre tareas se han respetado.

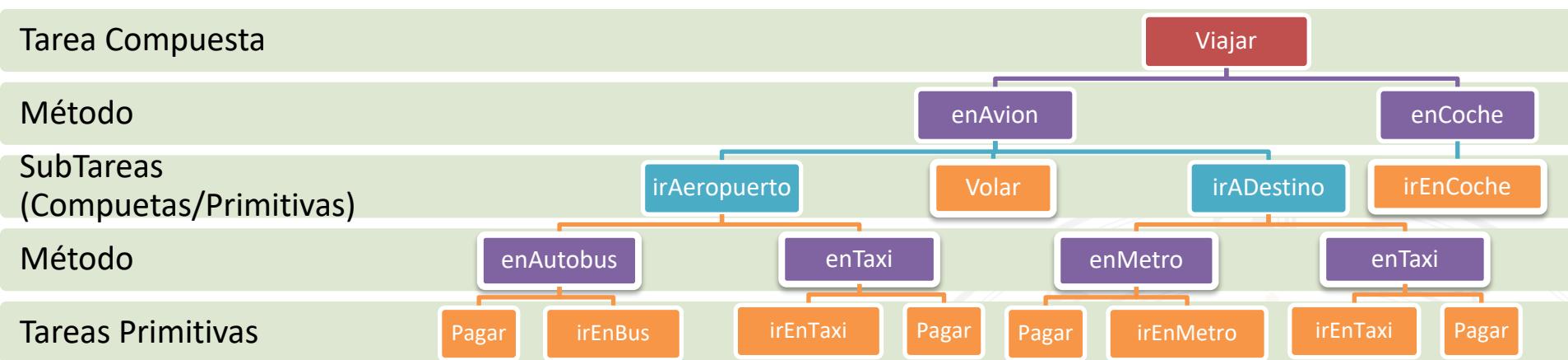


- Los operadores de reducción dan lugar a un espacio de búsqueda diferente al de planificación clásica:
 - Nodos: tareas/acciones
 - Arcos: conectan nodos relacionados con métodos de reducción
- Búsqueda en un grafo Y/O.



- Veamos un ejemplo de búsqueda primero en profundidad en un dominio de transporte multimodal.

Un dominio HTN de transporte multimodal



```

(:init
(en Yo MiCasa)
(= (dinero Yo) 100)
(= (distancia MiCasa GarciaLorca) 20)
(= (precio-km) 7)
)
(:tasks-goal
:tasks( (Viajar Yo MiCasa UnDestino) ))
  
```

Descomponer la Tarea de Nivel Superior (del problema)

(Viajar Yo
MiCasa
CasaMiPrimo)

Seleccionar un Método
(probando las precondiciones, en el orden en que se han escrito)

enAvion

enCoche

Descomponer la tarea siguiendo el orden de las subtareas

irAeropuerto

Volar

irADestino

Seleccionar un método

enTaxi

enBus

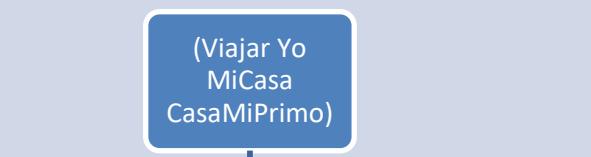
andando

Descomponer/aplicar si primitiva

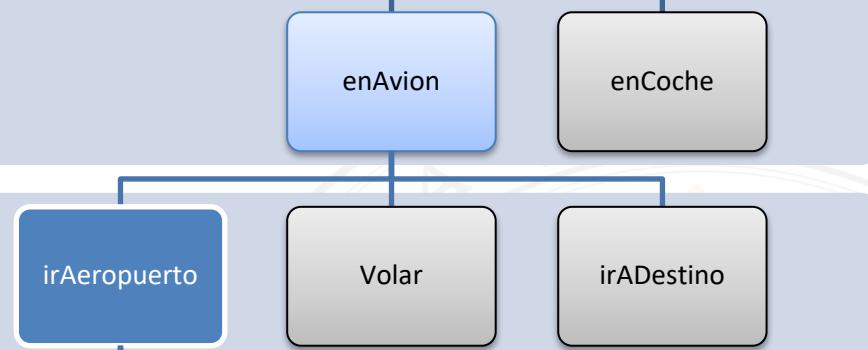
cogerTaxi



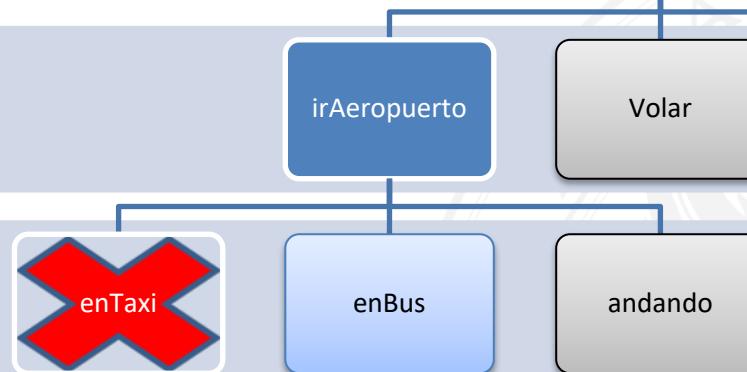
Descomponer la Tarea de Nivel Superior (del problema)



Seleccionar un Método
(probando las precondiciones, en el orden en que se han escrito)



Descomponer la tarea siguiendo el orden de las subtareas



Seleccionar un método



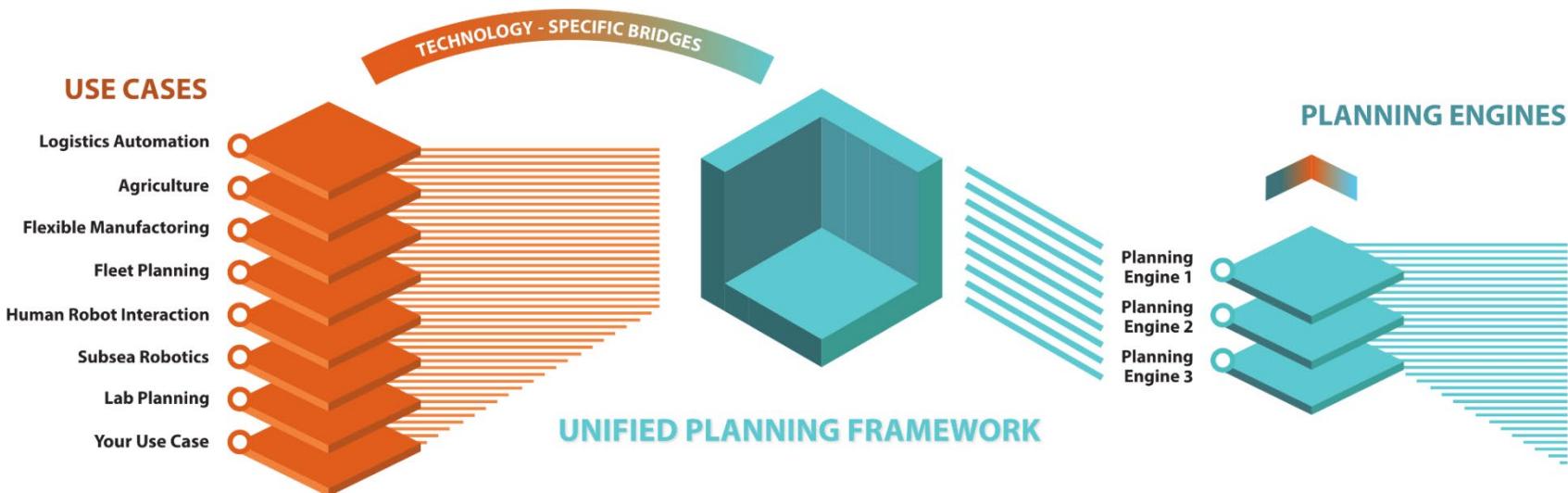
Descomponer/aplicar si primitiva

SIADEX es un planificador HTN integrado dentro del Proyecto “Unified Planning Framework”

<https://www.aiplan4eu-project.eu/>

<https://github.com/aiplan4eu/unified-planning>

<https://github.com/UGR-IntelligentSystemsGroup/up-siadex/tree/main>



- HPDL es una extensión de PDDL para representar dominios de planificación HTN basados en tareas primitivas y compuestas

- estándar para representar acciones de dominios de planificación

PDDL



- lenguaje desarrollado por el Grupo de Sistemas Inteligentes del Departamento de Ciencias de la Computación e I.A. (DECSAI)

HPDL



- Es el lenguaje usado por el planificador SIADEX también desarrollado por el grupo ISG

SIADEX



Conceptos

- **Objetos del dominio:**

- **constantes,**
- **tipos**
- **predicados,**
- **funciones**

Código

```
(define (domain viajes)
  (:requirements :typing :fluents      :derived-predicates
                 :negative-preconditions   :htn-expansion)
  (:constants <....>)
  (:types Persona Sitio - object
          Hogar Aeropuerto - Sitio)
  (:predicates (en ?p - Persona ?s - Sitio))
  (:functions (distancia ?x ?y - Sitio)
              (dinero ?p - Persona)
              (velocidad-taxi)
              (precio_km)
  )
)
```

- **Acciones primitivas (durativas):**

- Representación PDDL
- Parámetros con tipo,
- precondiciones,
- efectos,
- duración

```
(:durative-action ir_en_taxi
  :parameters (?u - Persona ?o ?d - Sitio)
  :duration (= ?dur (* (distancia ?o ?d)
                        (velocidad_taxi) ))
  :condition (and (en ?u ?o))
  :effect (and (not (en ?u ?o))
                (en ?u ?d)))
```

Expresiones aritméticas

- Valores numéricos/funciones
 - Especificar cómo calcular duraciones de acciones
 - Condiciones con expresiones aritméticas
- ```
(:durative-action ir_en_taxi
 :parameters (?u - Persona ?o ?d - Sitio)
 :duration (= ?dur (* (distancia ?o ?d)
 (velocidad_taxi)))
 :condition (and (en ?u ?o)
 (> (dinero ?u)
 (* (precio_km) (distancia ?o ?d)))
 :effect (and (not (en ?u ?o))
 (en ?u ?d))))
```

## Predicados derivados

- Valores numéricos/funciones
- Derived literals: reglas de inferencia para “derivar” predicados de las precondiciones

```
(:derived (tiene_dinero ?p - Persona ?org ?dst - Sitio)
 ((> (dinero ?p)
 (* (precio_km) (distancia ?org ?dst)))

(:durative-action ir_en_taxi
 :parameters (?u - Persona ?o ?d - Sitio)
 :duration (= ?dur (* (distancia ?o ?d)
 (velocidad_taxi))
 :condition (and (en ?u ?o)
 (tiene_dinero ?u ?o ?d))
 :effect (and (not (en ?u ?o))
 (en ?u ?d)))
```

## Operaciones sobre funciones.

- Valores numéricos/funciones
- Derived literals: reglas de inferencia para “derivar” predicados de las precondiciones
- Asignación, incremento/decremento de funciones

```
(:derived (tiene_dinero ?p - Persona ?org ?dst - Sitio)
 ((> (dinero ?u)
 (* (precio_km) (distancia ?o ?d))))
```

```
(:durative-action ir_en_taxi
:parameters (?u - Persona ?o ?d - Sitio)
:duration (= ?dur (* (distancia ?o ?d)
 (velocidad_taxi)))
:condition (and (en ?u ?o)
 (tiene_dinero ?u ?o ?d))
:effect (and (not (en ?u ?o))
 (en ?u ?d)))
```

```
(:durative-action pagar
:parameters (?u - Persona ?c - number)
:duration (= ?dur 1)
:condition (> (- (dinero ?u) ?c) 0)
:effect (decrease (dinero ?u) ?c)
```

## □ Tarea compuesta

## □ Varios métodos

- Precondición
- Descomposición

## □ Tareas “inline”

- Crear acciones “al vuelo” para un uso muy específico
- Inferir nuevo conocimiento y añadirlo al estado el mundo

## □ Relaciones de orden

- ( $<t_1> <t_2>$ )
- [ $<t_1> <t_2>$ ]

```
(:task irAeropuerto
:parameters (?p - Persona ?c - Hogar ?a - Aeropuerto)
(:method enTaxi
:precondition ()
:tasks (
(ir_en_taxi ?p ?c ?a)
(:inline (bind ?tarifa (* (distancia ?c ?a)
(precio_km))) ())

(pagar ?p ?tarifa)
)
(:method enBus
:precondition ()
:tasks ((:inline (bind ?tarifa (tarifa-bus)) ())
(pagar ?p ?tarifa)
(ir_en_bus ?p ?c ?a))
(:method Andando
:precondition ()
:tasks (irAndando ?p ?c ?a))
);;task
```

```
(:task irAeropuerto
:parameters (?p - Persona ?c - Hogar ?a - Aeropuerto)
(:method enTaxi
:precondition ()
:tasks (
(ir_en_taxi ?p ?c ?a)
(:inline (bind ?tarifa (* (distancia ?c ?a)
(precio_km))) ())
(pagar ?p ?tarifa)
))
(:method enBus
:precondition ()
:tasks ((:inline (bind ?tarifa (tarifa-bus)) ())
(pagar ?p ?tarifa)
(ir_en_bus ?p ?c ?a)))
(:method Andando
:precondition ()
:tasks (irAndando ?p ?c ?a)))
);;task
```

```
(:task Viajar
:parameters (?p - Persona
?x ?y - Sitio)
(:method enAvion
:precondition (tiene prisa ?p)
:tasks (
(irAeropuerto ?p ?x GarciaLorca)
(Volar ?p GarciaLorca Barajas)
(irDestino ?p Barajas ?y)
))
(:method enCoche
:precondition (not (tiene_prisa ?p))
:tasks (irEnCoche ?p ?x ?y)))
)
```

## Problema HTN:

- se describe a partir de un estado inicial y de un objetivo representado como una tarea de alto nivel a llevar a cabo.

```
(define (problem UnViaje) (:domain Viajes)

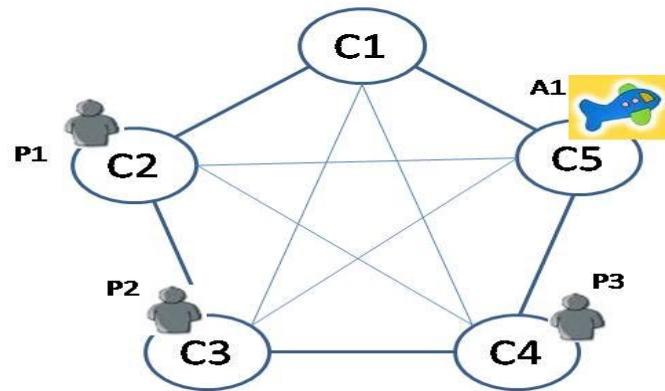
(:objects
 MiCasa CasaMiPrimo - Hogar
 GarciaLorca Barajas - Aeropuerto
 Yo - Persona
)
(:init
 (en Yo MiCasa)
 (= (dinero Yo) 100)
 (= (distancia MiCasa GarciaLorca) 20)
 (= (precio-km) 7)
)
(:tasks-goal
 :tasks((Viajar Yo MiCasa CasaMiPrimo))
)
```

Observar: el goal en HTN es una tarea, o una secuencia de tareas.

En PDDL (planificación clásica) se especificaría como un predicado (**en Yo CasaMiPrimo**)

- Definir un dominio HTN de forma incremental
  - No vamos a partir de cero para escribir este dominio.
  - Partimos de un conjunto de tareas primitivas (acciones PDDL) ya conocidas.
  - Tareas compuestas incompletas
  - El dominio final tiene que resolver, al menos, tres problemas
  - Resolver problemas en el dominio ZenoTravel (estándar para contrastar planificadores)
- Ejemplos de ficheros en el material de Prado.

- Transporte aéreo entre ciudades
- Personas, ciudades, aviones
- En concreto: 5 ciudades, 1 avión, 3 personas



- **Embarcar** una persona en un *avión* en una *ciudad* concreta, el embarque de una persona está sujeto a restricciones temporales
- **Desembarcar** una persona en un *avión* en una *ciudad* concreta, sujeto a restricciones temporales
- **Volar lento** un *avión* de una *ciudad origen* a una *ciudad destino* a una *velocidad lenta*, el *avión* tiene una **capacidad de recurso** (*fuel*) y tiene **consumo de recurso** (*fuel*) que depende de la *distancia recorrida* y del **ratio de consumo** a *velocidad lenta*
- **Volar rápido** un *avión* de una *ciudad origen* a una *ciudad destino* a una *velocidad rápida*, similares restricciones a *volar lento*.
- **Repostar** un *avión* en una *ciudad*, recarga la *capacidad del avión*.

# Preámbulo, tipos, predicados y funciones

```
(define (domain zeno-travel)
 (:requirements
 :typing
 :fluents
 :derived-predicates
 :negative-preconditions
 :universal-preconditions
 :disjunctive-preconditions
 :conditional-effects
 :htn-expansion
 ; Requisitos adicionales
 ; para el manejo del tiempo
 :durative-actions
 :metatags
)
```

- Este preámbulo debe respetarse tal cual.

# Preámbulo, tipos, predicados y funciones

(:types aircraft person city - object)

(:constants slow fast - object)

(:predicates

    (at ?x - (either person aircraft) ?c - city)

    (in ?p - person ?a - aircraft)

    (diferente ?x ?y) ;;predicado derivado, ver más abajo

    (igual ?x ?y) ;;predicado derivado, ver más abajo

    (hay-fuel ?a ?c1 ?c2) ;;predicado derivado, ver más abajo

    )

# Preámbulo, tipos, predicados y funciones

(:functions

```
(fuel ?a - aircraft) ;;cantidad de fuel actual de un avión
(distance ?c1 - city ?c2 - city) ;;distancia entre dos ciudades
(slow-speed ?a - aircraft); ;velocidad “Lenta” de un avión
(fast-speed ?a - aircraft) ;;velocidad “rápida” de un avión
(slow-burn ?a - aircraft);;razón de consumo de un avión a velocidad Lenta
(fast-burn ?a - aircraft);;razón de consumo de un avión a velocidad rápida
(capacity ?a - aircraft) ;;capacidad de fuel de un avión
(refuel-rate ?a - aircraft) ;;razón de repostaje de un avión (para calcular
;; el tiempo de repostaje
(total-fuel-used) ;;valor del fuel total usado
(boarding-time) ;;valor constante de tiempo de embarque
(debarking-time) ;;valor constante de tiempo de desembarque
)
```



# Embarcar una persona en un avión en una ciudad concreta.

```
(:durative-action board
:parameters (?p - person ?a - aircraft ?c - city)
:duration (= ?duration (boarding-time))
:condition (and (at ?p ?c)
 (at ?a ?c)))
:effect (and (not (at ?p ?c))
 (in ?p ?a)))
```



## Desembarcar una persona en un avión en una ciudad concreta.

```
(:durative-action debark
:parameters (?p - person ?a - aircraft ?c - city)
:duration (= ?duration (debarking-time))
:condition (and (in ?p ?a)
 (at ?a ?c)))
:effect (and (not (in ?p ?a))
 (at ?p ?c)))
```

# Volar un avión de una ciudad origen a una ciudad destino a una velocidad lenta

```
(:durative-action fly
:parameters (?a - aircraft ?c1 ?c2 - city)
:duration (= ?duration (/ (distance ?c1 ?c2) (slow-speed ?a)))
:condition (and (at ?a ?c1)
 (>= (fuel ?a)(* (distance ?c1 ?c2) (slow-burn ?a))))
:effect (and (not (at ?a ?c1))
 (at ?a ?c2)
 (increase (total-fuel-used)
 (* (distance ?c1 ?c2) (slow-burn ?a)))
 (decrease (fuel ?a)
 (* (distance ?c1 ?c2) (slow-burn ?a)))))
```

# Volar un avión de una ciudad origen a una ciudad destino a una velocidad rápida

```
(:durative-action zoom
:parameters (?a - aircraft ?c1 ?c2 - city)
:duration (= ?duration (/ (distance ?c1 ?c2) (fast-speed ?a)))
:condition (and (at ?a ?c1)
 (>= (fuel ?a) (* (distance ?c1 ?c2) (fast-burn ?a))))
:effect (and (not (at ?a ?c1))
 (at ?a ?c2)
 (increase (total-fuel-used)
 (* (distance ?c1 ?c2) (fast-burn ?a)))
 (decrease (fuel ?a)
 (* (distance ?c1 ?c2) (fast-burn ?a))))))
```

# Repostar un avión en una ciudad.

```
(:durative-action refuel
:parameters (?a - aircraft ?c - city)
:duration (= ?duration (/(- (capacity ?a) (fuel ?a))(refuel-rate ?a)))
:condition (and (> (capacity ?a) (fuel ?a))
(at ?a ?c))
:effect (assign (fuel ?a) (capacity ?a)))
```

Estas acciones primitivas están definidas en el fichero “Primitivas-Zenotravel.pddl”

- ;; el consecuente "vacío" se representa como "()" y significa "siempre verdad"
- ;;un objeto es siempre igual a sí mismo

(:derived

(igual ?x ?x) ())

- ;; dos objetos son diferentes si no son iguales

(:derived

(diferente ?x ?y)

{return ?x != ?y}

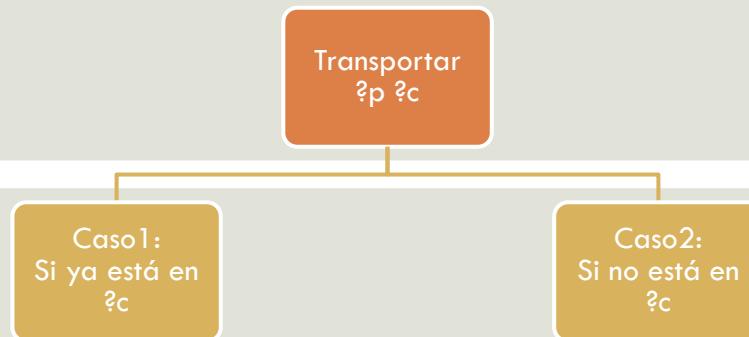
)

(:derived

(hay-fuel ?a - aircraft ?c1 - city ?c2 - city)

(> (fuel ?a) 1))

## Tarea Nivel Superior



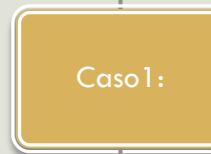
## Métodos

## Subtareas/Primitivas



## Métodos

## Subtareas/primitivas



# Transportar una persona a una ciudad destino

(:task transport-person

:parameters (?p - person ?c - city)

(:method Case1 ; si la persona esá en la ciudad no se hace nada

:precondition (at ?p ?c)

:tasks ()

)

;si la persona no está en la ciudad destino, pero avion y persona están en la misma ciudad

(:method Case2

:precondition (and (at ?p - person ?c1 - city)

(at ?a - aircraft ?c1 - city))

:tasks (

(board ?p ?a ?c1)

(mover-avion ?a ?c1 ?c)

(debark ?p ?a ?c )))

)



## Mover avión

```
(:task mover-avion
:parameters (?a - aircraft ?c1 - city ?c2 -city)
(:method fuel-suficiente
:precondition (hay-fuel ?a ?c1 ?c2)
:tasks (
 (fly ?a ?c1 ?c2)
)
)
)
```

- Comprobar que con este dominio básico se resuelve el problema siguiente:

```
(define (problem zeno-0)
 (:domain zeno-travel)
 (:customization
 (= :time-format "%d/%m/%Y %H:%M:%S")
 (= :time-horizon-relative 2500)
 (= :time-start "05/06/2007 08:00:00")
 (= :time-unit :hours))

```

```
(:objects
 p1 p2 p3 p4 - person
 c1 c2 c3 c4 c5 - city
 a1 - aircraft
)
```

```
(:init
 (at p1 c4)
 (at p2 c4)
 (at p3 c5)
 (at a1 c4))
```

```
(= (distance c1 c2) 100)
 (= (distance c2 c3) 100)
 (= (distance c3 c4) 100)
 (= (distance c4 c5) 100)
 (= (distance c5 c1) 100)
 (= (distance c1 c5) 100)
 (= (distance c1 c3) 150)
 (= (distance c1 c4) 150)
 (= (distance c2 c5) 150)
 (= (distance c2 c4) 150)
 (= (distance c3 c1) 150)
 (= (distance c3 c5) 150)
 (= (distance c4 c2) 150)
 (= (distance c4 c1) 150)
 (= (distance c5 c2) 150)
 (= (distance c5 c3) 150)
```

```
(= (fuel a1) 100000)
 (= (slow-speed a1) 10)
 (= (fast-speed a1) 20)
 (= (slow-burn a1) 1)
 (= (fast-burn a1) 2)
 (= (capacity a1) 100000)
 (= (refuel-rate a1) 1)
 (= (total-fuel-used) 0)
 (= (boarding-time) 1)
 (= (debarking-time) 1)
)
(:tasks-goal
 :tasks(
 (transport-person p1 c4)
 (transport-person p2 c5)
 (transport-person p3 c2))))
```

- Comprobar que con este dominio básico se resuelve el problema siguiente:

```
(define (problem zeno-0)
 (:domain zeno-travel)
 (:customization
 (= :time-format "%d/%m/%Y %H:%M:%S")
 (= :time-horizon-relative 2500)
 (= :time-start "05/06/2007 08:00:00")
 (= :time-unit :hours))

```

```
(:objects
 p1 p2 p3 p4 - person
 c1 c2 c3 c4 c5 - city
 a1 - aircraft
)
```

```
(:init
 (at p1 c4)
 (at p2 c4)
 (at p3 c5)
 (at a1 c4))
```

```
(= (distance c1 c2) 100)
 (= (distance c2 c3) 100)
 (= (distance c3 c4) 100)
 (= (distance c4 c5) 100)
 (= (distance c5 c1) 100)
 (= (distance c1 c5) 100)
 (= (distance c1 c3) 150)
 (= (distance c1 c4) 150)
 (= (distance c2 c5) 150)
 (= (distance c2 c4) 150)
 (= (distance c3 c1) 150)
 (= (distance c3 c5) 150)
 (= (distance c4 c2) 150)
 (= (distance c4 c1) 150)
 (= (distance c5 c2) 150)
 (= (distance c5 c3) 150)
```

```
(= (fuel a1) 100000)
 (= (slow-speed a1) 10)
 (= (fast-speed a1) 20)
 (= (slow-burn a1) 1)
 (= (fast-burn a1) 2)
 (= (capacity a1) 100000)
 (= (refuel-rate a1) 1)
 (= (total-fuel-used) 0)
 (= (boarding-time) 1)
 (= (debarking-time) 1)
)
(:tasks-goal
 :tasks(
 (transport-person p1 c4)
 (transport-person p2 c5)
 (transport-person p3 c2))))
```

## □ PROBLEMA 1: Comprobar que NO se resuelve el problema siguiente y modificar el dominio

|                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                      |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (define (problem zeno-0)<br>(:domain zeno-travel)<br>(:customization<br>(= :time-format "%d/%m/%Y %H:%M:%S")<br>(= :time-horizon-relative 2500)<br>(= :time-start "05/06/2007 08:00:00")<br>(= :time-unit :hours))<br><br>(:objects<br>p1 p2 p3 p4 - person<br>c1 c2 c3 c4 c5 - city<br>a1 - aircraft<br>)<br>(:init<br>(at p1 c4)<br>(at p2 c4)<br>(at p3 c5)<br>(at a1 c4) | (= (distance c1 c2) 100)<br>(= (distance c2 c3) 100)<br>(= (distance c3 c4) 100)<br>(= (distance c4 c5) 100)<br>(= (distance c5 c1) 100)<br>(= (distance c1 c5) 100)<br><br>(= (distance c1 c3) 150)<br>(= (distance c1 c4) 150)<br>(= (distance c2 c5) 150)<br>(= (distance c2 c4) 150)<br>(= (distance c3 c1) 150)<br>(= (distance c3 c5) 150)<br>(= (distance c4 c2) 150)<br>(= (distance c4 c1) 150)<br>(= (distance c5 c2) 150)<br>(= (distance c5 c3) 150) | (= (fuel a1) 100000)<br>(= (slow-speed a1) 10)<br>(= (fast-speed a1) 20)<br>(= (slow-burn a1) 1)<br>(= (fast-burn a1) 2)<br>(= (capacity a1) 100000)<br>(= (refuel-rate a1) 1)<br>(= (total-fuel-used) 0)<br>(= (boarding-time) 1)<br>(= (debarking-time) 1)<br>)<br><br>(:tasks-goal<br>:tasks(<br>(transport-person p1 c5)<br>(transport-person p2 c5)<br>(transport-person p3 c5) |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

## □ PROBLEMA 2: Comprobar que NO se resuelve el problema siguiente y volver a modificar el dominio

```
(define (problem zeno-0)
 (:domain zeno-travel)
 (:customization
 (= :time-format "%d/%m/%Y %H:%M:%S")
 (= :time-horizon-relative 2500)
 (= :time-start "05/06/2007 08:00:00")
 (= :time-unit :hours))
 (:objects
 p1 p2 p3 p4 - person
 c1 c2 c3 c4 c5 - city
 a1 - aircraft
)
 (:init
 (at p1 c4)
 (at p2 c4)
 (at p3 c5)
 (at a1 c4)
))
```

|                          |                          |
|--------------------------|--------------------------|
| (= (distance c1 c2) 100) | (= (fuel a1) 200)        |
| (= (distance c2 c3) 100) | (= (slow-speed a1) 10)   |
| (= (distance c3 c4) 100) | (= (fast-speed a1) 20)   |
| (= (distance c4 c5) 100) | (= (slow-burn a1) 1)     |
| (= (distance c5 c1) 100) | (= (fast-burn a1) 2)     |
| (= (distance c1 c5) 100) | (= (capacity a1) 300)    |
|                          | (= (refuel-rate a1) 1)   |
| (= (distance c1 c3) 150) | (= (total-fuel-used) 0)  |
| (= (distance c1 c4) 150) | (= (boarding-time) 1)    |
| (= (distance c2 c5) 150) | (= (debarking-time) 1)   |
| (= (distance c2 c4) 150) | )                        |
| (= (distance c3 c1) 150) | (:tasks-goal             |
| (= (distance c3 c5) 150) | :tasks(                  |
| (= (distance c4 c2) 150) | (transport-person p1 c5) |
| (= (distance c4 c1) 150) | (transport-person p2 c5) |
| (= (distance c5 c2) 150) | (transport-person p3 c5) |
| (= (distance c5 c3) 150) |                          |

□ PROBLEMA •3: Comprobar que NO se resuelve el problema siguiente y modificar el dominio

|                                                                                                                                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |                                                                                                                                                                                                                                                                                                                                                                                                         |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| (define (problem zeno-0)<br>(:domain zeno-travel)<br>(:customization<br>(= :time-format "%d/%m/%Y %H:%M:%S")<br>(= :time-horizon-relative 2500)<br>(= :time-start "05/06/2007 08:00:00")<br>(= :time-unit :hours))<br><br>(:objects<br>p1 p2 p3 p4 - person<br>c1 c2 c3 c4 c5 - city<br>a1 - aircraft<br>)<br>(:init<br>(at p1 c4)<br>(at p2 c4)<br>(at p3 c5)<br>(at a1 c4) | (= (distance c1 c2) 100)<br>(= (distance c2 c3) 100)<br>(= (distance c3 c4) 100)<br>(= (distance c4 c5) 100)<br>(= (distance c5 c1) 100)<br>(= (distance c1 c5) 100)<br><br>(= (distance c1 c3) 150)<br>(= (distance c1 c4) 150)<br>(= (distance c2 c5) 150)<br>(= (distance c2 c4) 150)<br>(= (distance c3 c1) 150)<br>(= (distance c3 c5) 150)<br>(= (distance c4 c2) 150)<br>(= (distance c4 c1) 150)<br>(= (distance c5 c2) 150)<br>(= (distance c5 c3) 150) | (= (fuel-limit) 1500)<br>(= (fuel a1) 200)<br>(= (slow-speed a1) 10)<br>(= (fast-speed a1) 20)<br>(= (slow-burn a1) 1)<br>(= (fast-burn a1) 2)<br>(= (capacity a1) 300)<br><br>(= (refuel-rate a1) 1)<br>(= (total-fuel-used) 0)<br>(= (boarding-time) 1)<br>(= (debarking-time) 1)<br>)<br>(:tasks-goal<br>:tasks(<br>(transport-person p1 c5)<br>(transport-person p2 c5)<br>(transport-person p3 c5) |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

- Planificación clásica: resolución de problemas clásica.

- Estado inicial: conjunto de predicados instanciados
- Operadores: acciones
- Objetivo: conjunto de predicados instanciados
- Proceso de búsqueda:
  - Espacio de estados:
    - *operadores espacio búsqueda == acciones*
    - *generación de estados sucesores y se para cuando encontramos un camino desde el estado inicial hasta el objetivo.*
    - *devuelve el camino*
  - Espacio de planes:
    - *operadores espacio búsqueda == operaciones de refinamiento de un plan inicial*
    - *generación de planes sucesores aplicando operadores de refinamiento y se para cuando encontramos un plan que ya no se puede refine más.*
    - *devuelve el plan completamente refinado*
- Plan: el camino desde el estado inicial hasta el objetivo.

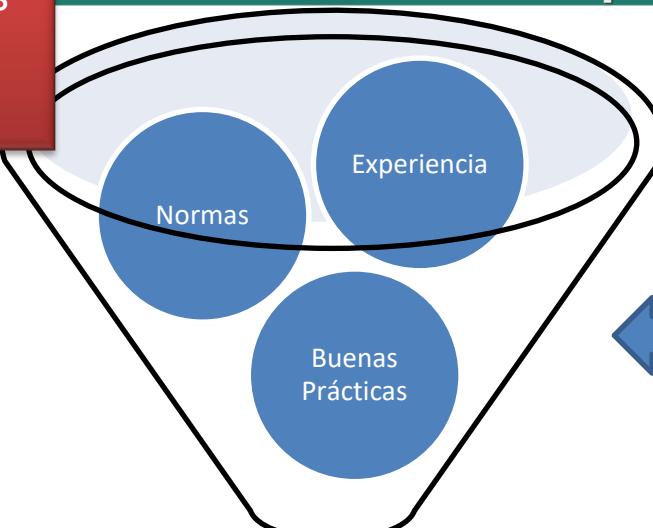
- Planificación jerárquica: resolución de problemas usando conocimiento experto.

- Estado inicial: conjunto de predicados instanciados
- **Nuevos conceptos:** tarea y red de tareas
  - Dos tipos de tareas: compuestas y primitivas.
  - Red de tareas: una secuencia (parcialmente) ordenada de tareas (primitivas o compuestas)
- Operadores, dos tipos:
  - Primitivas (equivalente a acciones clásicas)
  - métodos de descomposición: definen formas alternativas de descomponer una tarea.
- Objetivo:
  - Red de tareas inicial
- Proceso de búsqueda: espacio de redes de tareas.
  - operadores: métodos de descomposición
  - reducción recursiva de tareas iniciales generando un arbol (o bosque) de descomposición hasta encontrar una red de tareas sólo primitivas.
- Plan: una red de tareas primitivas.

¿En qué marco se desarrollan aplicaciones de planificación jerárquica?

# Planificación HTN EN LA PRÁCTICA

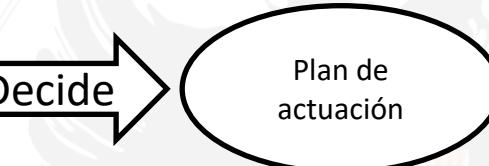
Modelos de proceso con conocimiento experto



Conocimiento



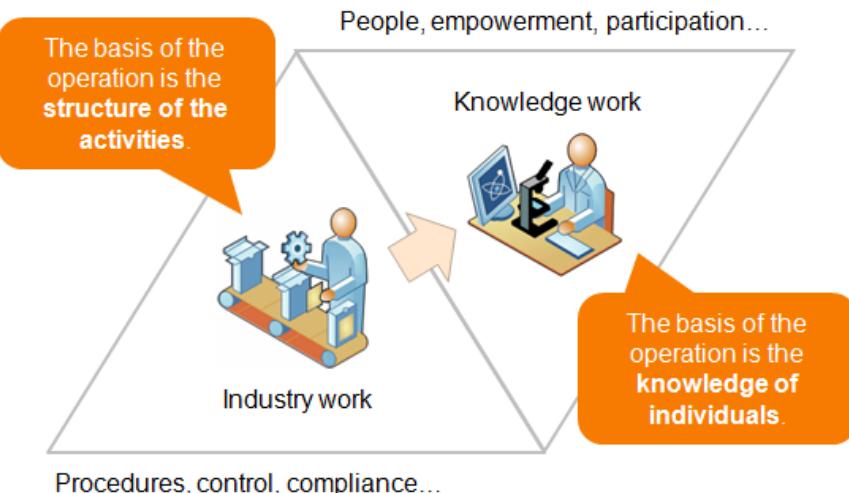
Decide



Los ejemplos de aplicación comentados(medicina, educación, gestión de emergencias, turismo electrónico) son en realidad ejemplos de lo que se ha venido a llamar trabajo basado en el conocimiento o *knowledge work*. Los **modelos de proceso usados en BPMN**, en su mayor parte representados en BPMN no nos sirven para representar este conocimiento experto porque están **limitados a modelar el trabajo rutinario**

- **Un tipo de trabajo bien conocido a priori**
  - Puede planificarse a un buen nivel de detalle
  - Puede hacerse de forma muy similar cada vez
- **Predecible y repetible,**
  - por tanto puede ser automatizado por medios tradicionales de automatización de procesos.
- **Ejemplo:**
  - Cualquier procedimiento de operario de banca, seguros ...
  - No es que se haga mecánicamente, exactamente igual cada vez.
  - Pero hay suficiente similaridad en cada instancia de trabajo
- **Hay beneficio en identificar un patrón de trabajo específico, detallado .**
  - ¿Hasta qué detalle puede planificarse el tratamiento de un enfermo, el plan de extinción de un incendio, el itinerario de un alumno o un plan de visitas?
  - ¿En qué medida son predecibles y repetibles?.

Los procesos de los que hemos hablado al principio de la sesión son **procesos orientados a objetivos** y para los que se necesita modelar "trabajo del conocimiento" o **knowledge work** y dar soporte a las decisiones y tareas diarias del "trabajador del conocimiento" o **knowledge worker**.



- Knowledge work no se parece a los procesos de negocio tradicionales
  - Porque conforme el proceso avanza la secuencia de acciones depende mucho sobre lo específico de la situación
    - P.ej: quién y qué está disponible y qué opciones particulares existen en ese momento
  - Una parte de hacer ese trabajo es hacer el plan en sí mismo.
    - El plan inicial puede hacerse con cualquier información disponible en ese momento, pero es necesario al menos un plan inicial.
  - Puede incorporar elementos de procedimiento previamente preparados.
    - Por ejemplo: hay protocolos que deben seguirse para realizar test de laboratorio
- ....

Swenson, Keith D. *Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done*. Tampa, FL: Meghan-Kiffer Press, 2010.

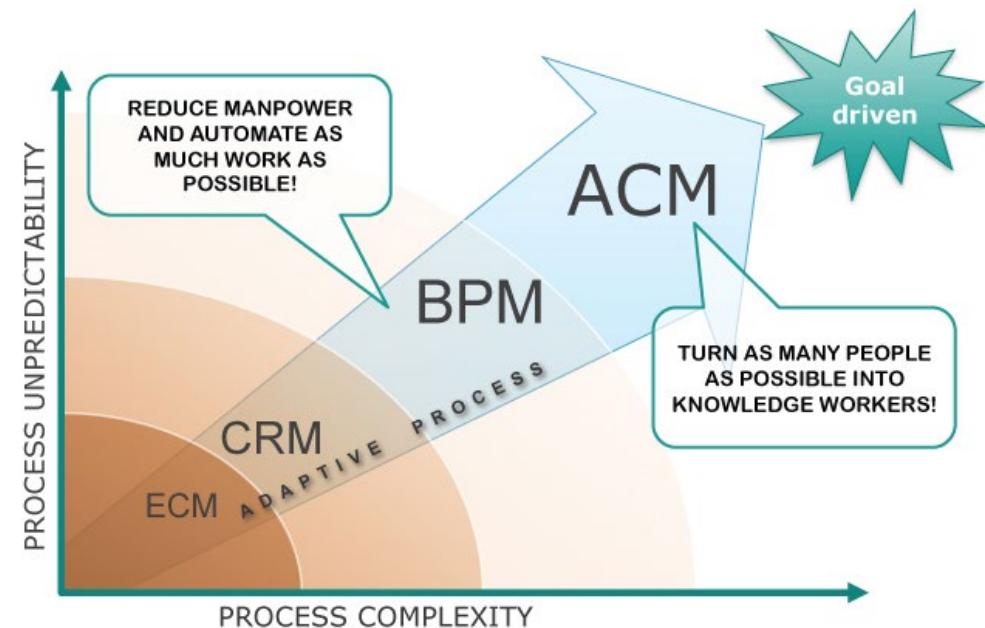
En los procesos de *knowledge work* la información sobre la situación cambia constantemente (**entorno dinámico**), las actividades no tienen resultados completamente predecibles (**conocimiento incompleto**) y los objetivos pueden necesitar reformularse. Por tanto el plan requiere un ciclo continuo de evaluar, planificar y ejecutar. No hay un plan final hasta que el objetivo se ha alcanzado.



ACM es un sector de negocio surgido para dar respuesta a los requisitos del trabajo del conocimiento

Etiquetas: comerciales y académicas

- Adaptive Case Management
- Dynamic Case Management
- Smart Process Apps
- Intelligent BPM
- Flexible BPM



## Procesos Dinámicos:

- dependen de una situación dada (context dependent) y responden ante cambios de situaciones.

## Flexibles:

- no rígidos (obvio), pueden cambiarse mientras se usan.

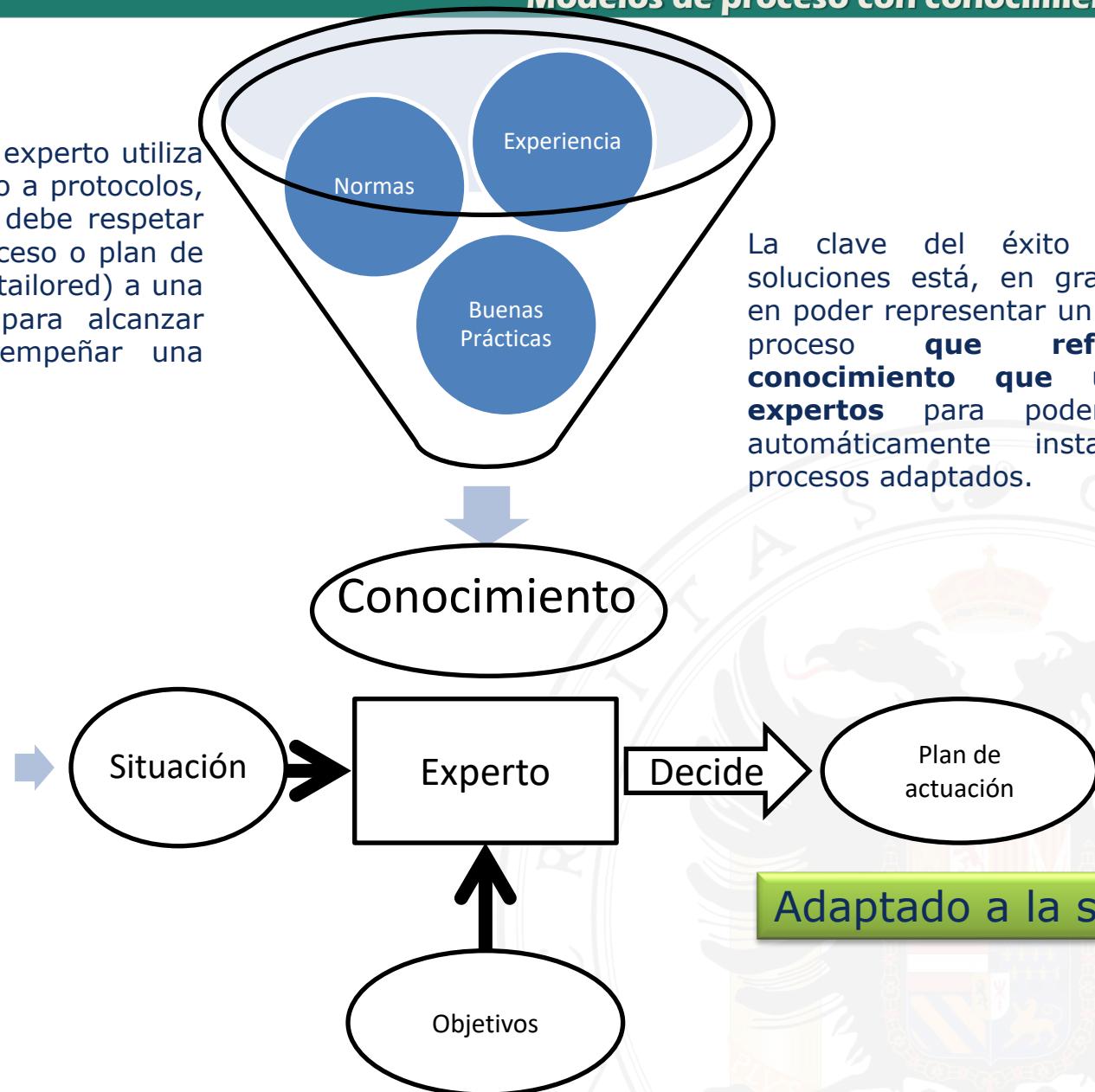
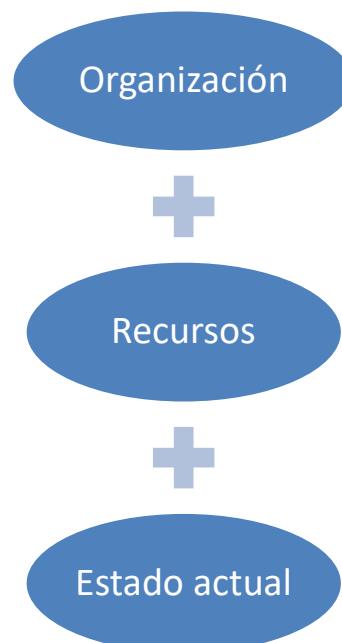
## Adaptativos:

- automáticamente producen cambios internos causados por condiciones externas, ajustándose a las nuevas condiciones.

## Ágiles:

- se adaptan rápidamente y sin mucha preparación o soporte.

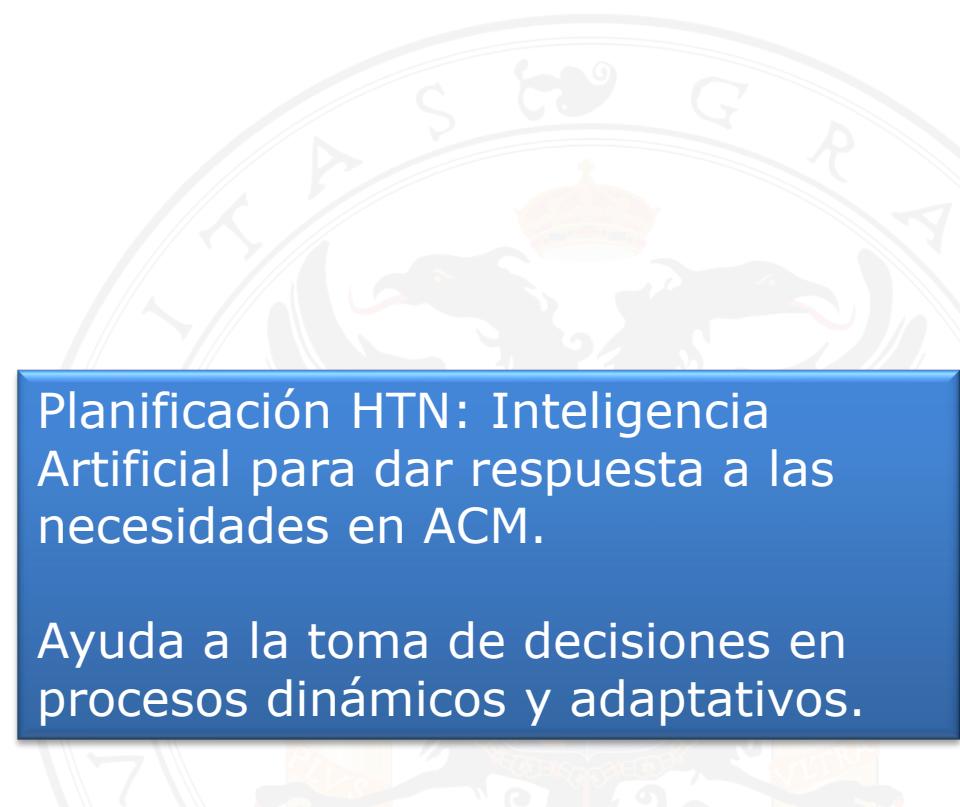
En estos ejemplos el experto utiliza su conocimiento junto a protocolos, normas o guías que debe respetar para elaborar un proceso o plan de actuación adaptado (tailored) a una situación concreta, para alcanzar un objetivo o desempeñar una tarea concreta.



La clave del éxito en estas soluciones está, en gran medida, en poder representar un modelo de proceso **que refleje el conocimiento que usan los expertos** para poder generar automáticamente instancias de procesos adaptados.

• Ingredientes para entender cómo generar procesos orientados a objetivos que dependen de una situación.

- Modelo de proceso, basado en un dominio de planificación HTN.
- Generación dinámica, basada en un planificador HTN.
- Entradas: Objetivo HTN y estado inicial representando propiedades y relaciones entre objetos.
- Salidas: un plan propuesto para alcanzar un objetivo o desempeñar una tarea.



Planificación HTN: Inteligencia Artificial para dar respuesta a las necesidades en ACM.

Ayuda a la toma de decisiones en procesos dinámicos y adaptativos.

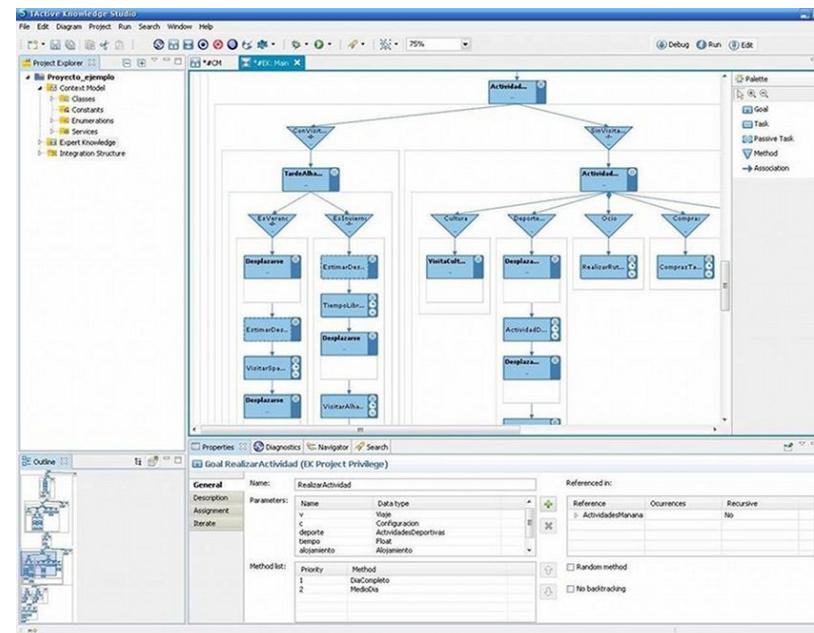
- HPDL:
  - Lenguaje extensión HTN con información temporal

L. Castillo, J. Fdez-Olivares, O. García-Pérez, y F. Palao, «Efficiently handling temporal knowledge in an HTN planner», en *16th International Conference on Automated Planning and Scheduling (ICAPS-06)*, 2006, pp. 63–72.

- Expresivo para representar modelos de proceso con conocimiento experto humano:
  - Workflow patters comunes en BPM
  - Representación y gestión de recursos.
  - Restricciones temporales

J. Fdez-Olivares, L. Castillo, J. A. Cózar, y O. García Pérez, «Supporting clinical processes and decisions by hierarchical planning and scheduling», *Computational Intelligence*, vol. 27, n.º 1, pp. 103–122, 2011.

- HPDL es un lenguaje textual, usado para investigación y experimentación tecnológica.
- Existe una notación gráfica, **EKDL**, soportada por un entorno de desarrollo visual, desarrollado por IActive Intelligent Technologies.
- Sesión Práctica 2.



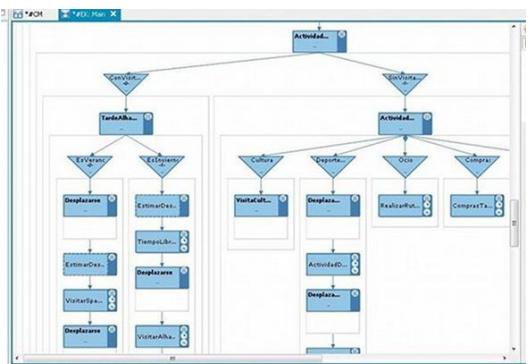
F. Palao, J. Fdez-Olivares, L. Castillo, y O. García, «An extended HTN knowledge representation based on a graphical notation», en *Proceedings of the Workshop on Knowledge Engineering for Planning and Scheduling (KEPS 2011)*, Freiburg, Germany, 2011, pp. 126-135.



## Dominio



## Objetivos



**Knowledge Studio**  
Suite Gráfica



Ingeniero  
Conocimiento

## Contexto

|    | PersonID | Address                  | Qty  | Op  | Firmante       | FolioRes | Lst   |
|----|----------|--------------------------|------|-----|----------------|----------|-------|
| 1  | 1        | 100 East Street          | 100  | Get | Hector Q.      | Loreto   | Div   |
| 2  | 2        | 100 West Street          | 100  | Get | Hector Q.      | Loreto   | Div   |
| 3  | 3        | 992 Tennessee Avenue     | 992  | Get | Hector Q.      | Loreto   | Div   |
| 4  | 4        | 780 Fourth Lane          | 780  | Get | Tenderlon      | Chad     | Mc    |
| 5  | 5        | 100 North Lane           | 100  | Get | Tenderlon      | Chad     | Mc    |
| 6  | 6        | 730 North Lane           | 730  | Get | Tenderlon      | Chad     | Mc    |
| 7  | 7        | 978 Eighth Walk          | 978  | Get | Melissa Park   | Pearla   | Rico  |
| 8  | 8        | 247 Fifth Place          | 247  | Get | Melissa Park   | Pearla   | Rico  |
| 9  | 9        | 345 Sixth Street         | 345  | Get | Melissa Park   | Pearla   | Rico  |
| 10 | 10       | 249 Washington Street    | 249  | Get | Oscar Center   | Oscar    | Gut   |
| 11 | 11       | 360 Tennessee Place      | 360  | Get | Oscar Center   | Oscar    | Gut   |
| 12 | 12       | 456 South Street         | 456  | Get | Oscar Center   | Oscar    | Gut   |
| 13 | 13       | 888 Love Avenue          | 888  | Get | Domingo Wright | Kathleen | Qui   |
| 14 | 14       | 698 Third Street         | 698  | Get | Domingo Wright | Kathleen | Qui   |
| 15 | 15       | 229 Kentucky Place       | 229  | Get | Domingo Wright | Kathleen | Qui   |
| 16 | 16       | 1019 Haven Gardens Place | 1019 | Get | Peter Hill     | Steve    | Steve |

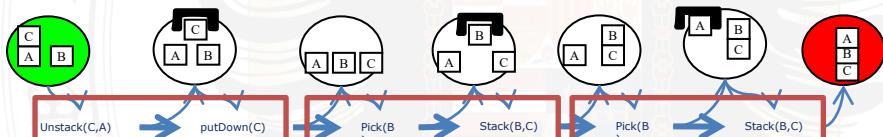
**Knowledge Studio**

Integración

**Decisor**  
HTN Planner



Bases de datos  
Externas

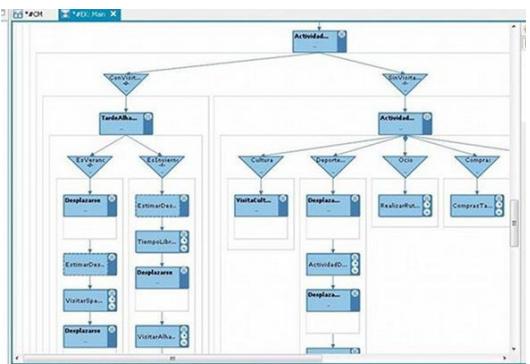




## Dominio



## Objetivos



**Knowledge Studio**  
Suite Gráfica



Ingeniero  
Conocimiento

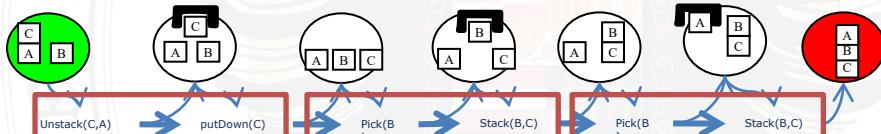
## Contexto

|    | PersonID | Address                  | Qty  | Op        | Firmante         | FolioRes         | Lst    |
|----|----------|--------------------------|------|-----------|------------------|------------------|--------|
| 1  | 1        | 1040 East Street         | 100  | Get Sheet | Helen Q.         | Helen Q.         | Div    |
| 2  | 2        | 124 North Park           | 120  | Get Sheet | Harold           | Harold           | Div    |
| 3  | 3        | 952 Tennessee Avenue     | 952  | Get Sheet | Harold           | Harold           | Div    |
| 4  | 4        | 250 Fourth Lane          | 760  | Get Sheet | Ted              | Ted              | Div    |
| 5  | 5        | 103 South Hwy            | 103  | Get Sheet | Carol            | Carol            | Div    |
| 6  | 6        | 730 North Lane           | 730  | Get Sheet | Karen            | Karen            | Rio    |
| 7  | 7        | 575 Eighth York          | 722  | Get Sheet | Melissa Park     | Melissa Park     | New    |
| 8  | 8        | 247 Fifth Place          | 247  | Get Sheet | Victoria Addison | Victoria Addison | Div    |
| 9  | 9        | 345 South Blvd           | 345  | Get Sheet | Steve            | Steve            | Div    |
| 10 | 10       | 249 Washington Street    | 745  | Get Sheet | Olivia Center    | Olivia Center    | Olivia |
| 11 | 11       | 360 Tennessee Place      | 360  | Get Sheet | Father's Name    | Father's Name    | Law    |
| 12 | 12       | 145 South Blvd           | 145  | Get Sheet | David            | David            | Div    |
| 13 | 13       | 888 Love Avenue          | 660  | Get Sheet | Dawn             | Dawn             | John   |
| 14 | 14       | 898 Third Street         | 898  | Get Sheet | Oscar Center     | Oscar Center     | Beth   |
| 15 | 15       | 229 Kentucky Place       | 229  | Get Sheet | Ocean View       | Ocean View       | Kim    |
| 16 | 16       | 1012 Marin Gardens Place | 1012 | Get Sheet | Peter Hill       | Peter Hill       | Steve  |



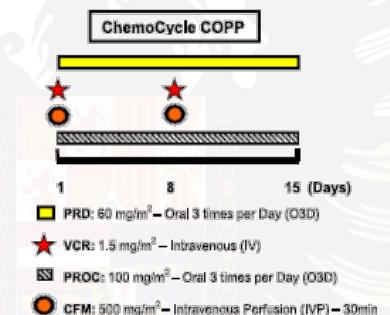
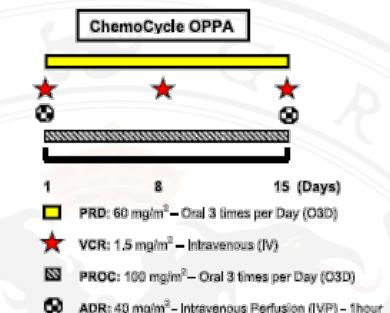
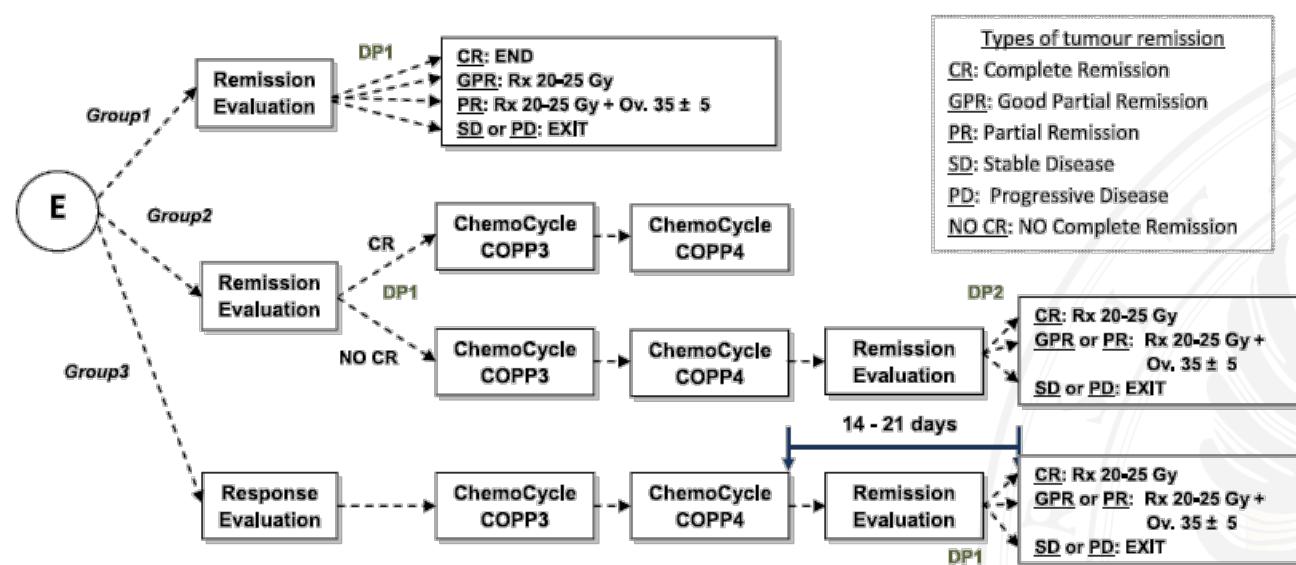
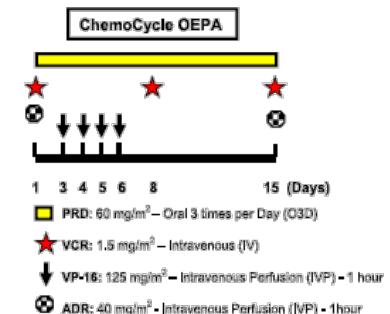
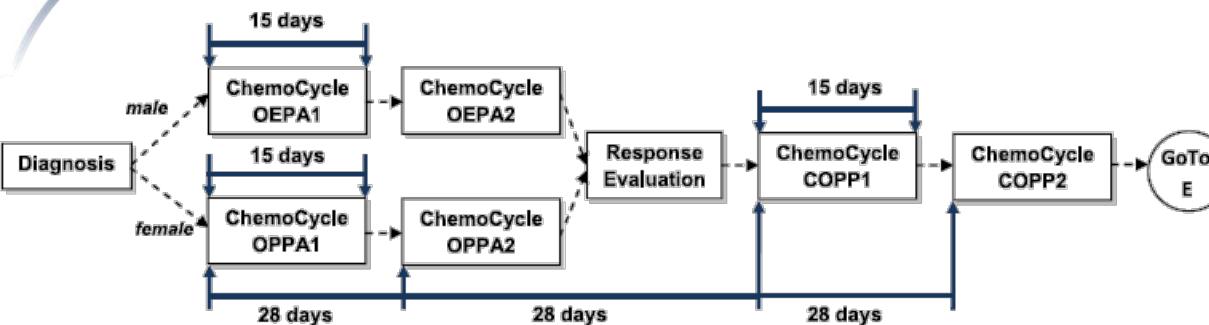
**Knowledge Studio**

Integración



HELLO!





- Ejemplo HPDL usado para
  - representar un protocolo clínico en oncología pediátrica.
  - representar estado del paciente, recursos del hospital
  - representar las tareas objetivo a cumplir para tratar al paciente.



Una enfermera administra un fármaco a un paciente con una dosis y modo de administración determinados

```
(:durative-action AdminDrug
 :parameters (?p - Patient ?n - Nurse ?d - Drug ?dose - number ?mode - Mode
 ?cycle - Cycle ?dur - number)
 :meta (
 (:tag prettyprint "The nurse ?n Administers the drug ?d to the patient ?p
 this dose ?dose and this mode ?mode. The current chemot
 cycle is ?cycle")
 (:tag short "AdminDrug ?d")
 (:tag resource "?n"))
 :duration (= ?duration ?dur)
 :condition (and (> (leukocytes ?p) 2000) (> (neutrophils ?p) 500)
 (> (platelets ?p) 100000) (available ?n))
 :effect (and (increase (total_dose ?p ?d) ?dose)
 (assign (begin-last ?d) ?start)))
)
```

Variables para restricciones temporales  
?start, ?end, ?dur

Meta información:  
• Texto informativo  
• ?n es un recurso

Una enfermera administra un fármaco a un paciente con una dosis y modo de administración determinados

```
(:durative-action AdminDrug
 :parameters (?p - Patient ?n - Nurse ?d - Drug ?dose - number ?mode - Mode
 ?cycle - Cycle ?dur - number)
 :meta (
 (:tag prettyprint "The nurse ?n Administers the drug ?d to the patient ?p
 this dose ?dose and this mode ?mode. The current chemot
 cycle is ?cycle")
 (:tag short "AdminDrug ?d")
 (:tag resource "?n"))
 duration (= ?duration ?dur)
 condition (and (> (leukocytes ?p) 2000) (> (neutrophils ?p) 500)
 (> (platelets ?p) 100000) (available ?n))
 effect (and (increase (total_dose ?p ?d) ?dose)
 (assign (begin-last ?d) ?start)))
)
```

El estado del paciente es OK y la enfermera está disponible.

Cambio de estado

Se incrementa la dosis total. El inicio de esta acción se guarda en otra variable temporal para usarla en otras restricciones temporales

```
(: task ClinicalProtocol
:parameters (?p - Patient ?date - Date)
(: method Man
:precondition (= (gender ?p) Man)
:tasks(
 ((start >= ?date) (PreEval ?p))
 [(AdministerCycle ?p VCR OEPA)
 (AdministerCycle ?p ADR OEPA)])
))
(: method Woman
:precondition(= (gender ?p) Woman)
:tasks(
 ((start >= ?date) (PreEval ?p))
 [(AdministerCycle ?p PRD OPPA)
 (AdministerCycle ?p PROC OPPA)])
))
```

### Tarea aplicar protocolo

#### Método alternativo para varón

##### Procedimiento:

1. Evaluar paciente
2. Aplicar 2 ciclos en paralelo

#### Restricción temporal en tareas.

#### Método alternativo para hembra

Workflow  
Patterns

Condicional

Sequence  
Parallel

```
(: task AdministerCycle
:parameters (?p - Patient ?d - Drug ?c - Cycle)
(: method repeat
:precondition (> (NRep ?d) 0)
:tasks(
(:inline () (decrease (NRep ?d) 1))
(:inline () (assign ?do
(* (bsa ?p) (dose ?d))))
(:inline () (assign ?m (mode ?d)))
(:inline () (assign ?dur (dur ?d)))
((= ?start (+ (begin-last ?d) 1440))
(AdminDrug ?p ?n ?d ?do ?m ?c ?dur))
(AdministerCycle ?p ?d ?c)))
(: method exit
:precondition (= (NRep ?d) 0)
:tasks()))
)
```

Tarea repetitiva aplicar ciclo quimio

Recurrir mientras NRep > 0

Tareas para inferencia interna

Restricción temporal para sincronizar con tareas previas.

Descomposición recursiva

Método para caso base recursión

Workflow Patterns

Cyclical

Patient Profile

```
;;definition of predicates in the domain
(sex ?patient ?s - gender)
(group ?patient ?g - group)
```

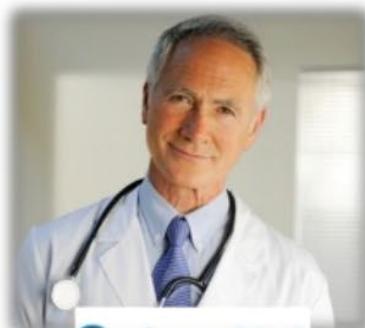
```
;; instances for patient Alice in the problem
(sex Alice M)
(group Alice Group3)
;;start date for treatment
(startdate Alice "07/11/2011 08:00:00")
```

```
(:tasks-goal
 :tasks ((Diagnosis Juan)
 (Treatment Juan)
 (Follow-up Juan)
)
```

```
(between "07/11/2011 00:00:00" and "07/12/2011 00:00:00" (available John))
(between "07/12/2011 00:00:00" and "07/01/2011 00:00:00" (available Paul))
(between "07/01/2012 00:00:00" and "07/02/2012 00:00:00" (available John))
(between "07/02/2012 00:00:00" and "07/03/2012 00:00:00" (available Paul))
(between "07/03/2012 00:00:00" and "07/04/2012 00:00:00" (available John))
```

Resource Constraints

## Cognocare is based on IActive's award-wining technology



Gartner 2012  
Cool Vendor

"How Knowledge Workers Get Things Done:  
Real-World Adaptive Case Management", 2012



### Award-winning Artificial Intelligence engine



Global Awards for Excellence in  
Adaptive Case Management. Gold  
Winner of the Healthcare category.  
Workflow Management Coalition  
2012, USA.



International Conference on  
Planning & Scheduling. Award for  
*Excellence in Knowledge  
Engineering.*  
ICAPS 2009. Tesalónica, Greece.

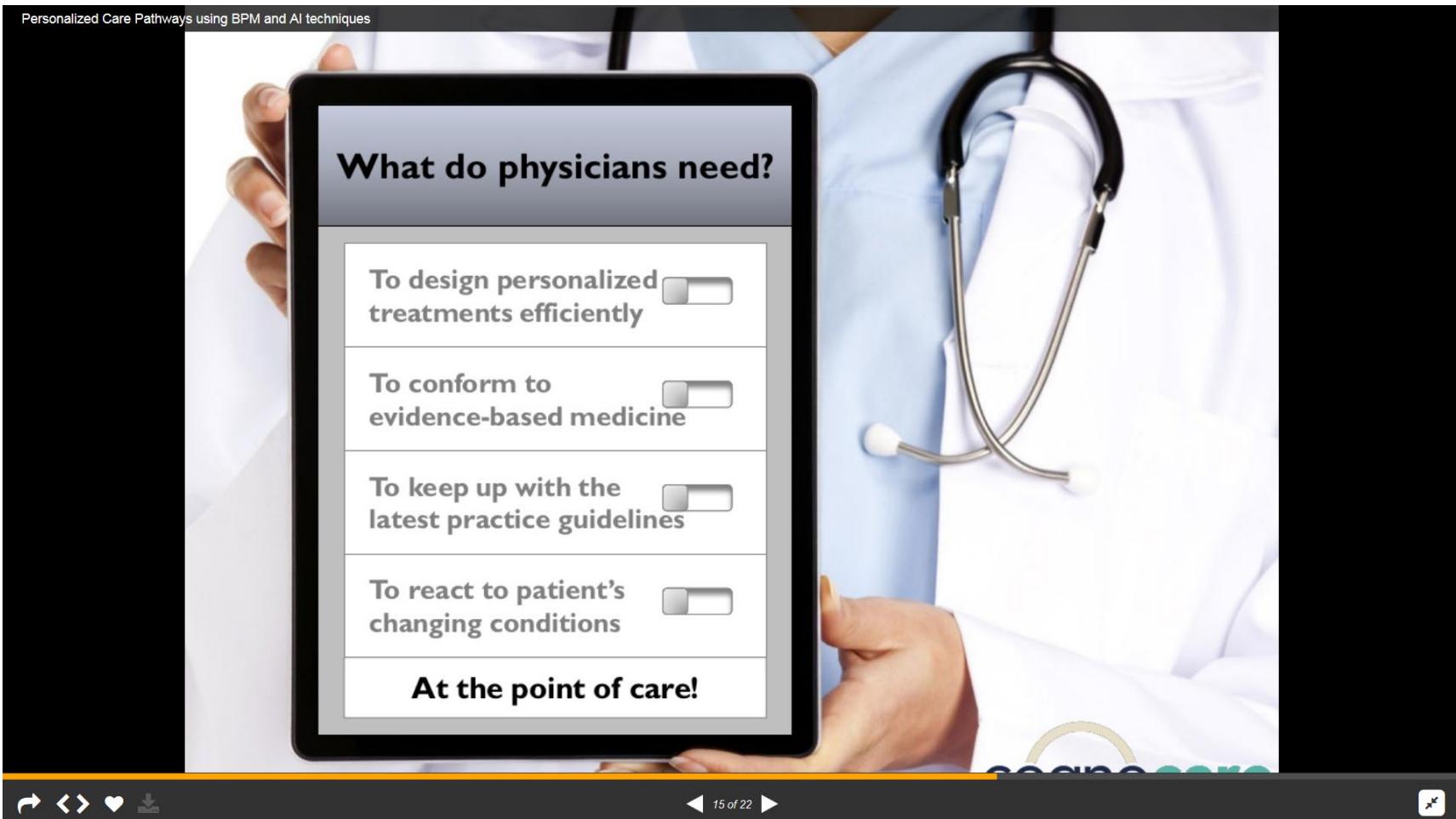


Spain National Informatics Congress.  
*Best Application Using Artificial  
Intelligence.*  
CEDI 2005, Spain.



International Conference on  
Planning & Scheduling. Award for  
*Best Application.*  
ICAPS 2006. United Kingdom.

Personalized Care Pathways using BPM and AI techniques



What do physicians need?

- To design personalized treatments efficiently
- To conform to evidence-based medicine
- To keep up with the latest practice guidelines
- To react to patient's changing conditions

**At the point of care!**

◀ 15 of 22 ▶

Personalized Care Pathways using BPM and AI techniques  
Physicians may modify the details of the treatment || Alerts about scheduled lab tests || Detailed explanations about dosages || Next scheduled test

Luis Castillo 4DEMO | MRN: 1234567 | Age: 4 years | Protocol: LAL | Group: STANDARD-RISK | Cycle: INDUCTION - week 2

**Cycle View** 

**INDUCTION** [CONSOLIDATION >](#)

\* Make pharmacogenetic studio for administration MTX in the consolidation cycle [Delete](#) [Remember later](#)

|                  | 6       | 7       | 8       | 9       | 10      | 11      | 12      | 13      | 14      | 15      | 16      | 17      | 18      | 19      | 20      | 21      | 22      | 23      | 24      | 25       | 26       | 27       | 28       | 29       | 30       | 1        | 2        | 3        | 4        | 5        | 6        | 7 |  |  |
|------------------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|---------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|----------|---|--|--|
| Daunorubicin     | 1.5 mg  | 1.5 mg  |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |          |          |          |          |          |          |          |          |          |          |          |          |   |  |  |
| Vincristine      | 1.5 mg  |         |         |         |         |         |         |         |         | 1.5 mg  |         |         |         |         |         |         |         |         |         |          |          |          |          |          |          |          |          |          |          |          |          |   |  |  |
| Prednisone       | 2 doses  | 2 doses  | 2 doses  | 2 doses  | 2 doses  | 2 doses  | 2 doses  | 2 doses  | 2 doses  | 2 doses  | 2 doses  | 2 doses  |   |  |  |
| Asparaginase     |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         | <300.0 u |   |  |  |
| Triple IT        | 3 Dose  |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |          |          |          |          |          |          |          |          |          |          |          |          |   |  |  |
| Cyclophosphamide |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         | 860.0 mg |          |          |          |          |          |          |          |          |          |          |          |   |  |  |
| TESTS            |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |          |          |          |          |          |          |          |          |          |          |          |          |   |  |  |
| AIDS             |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |          |          |          |          |          |          |          |          |          |          |          |          |   |  |  |
| RADIO THERAPY    |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |         |          |          |          |          |          |          |          |          |          |          |          |          |   |  |  |

**Supplies Drugs on Cycle** 

| Drug         | Scheduled | Supplied  | Canceled | Pending |
|--------------|-----------|-----------|----------|---------|
| Daunorubicin | 103.20 mg | 103.20 mg | 0.00 mg  | 0.00 mg |
| Vincristine  | 5.16 mg   | 2.58 mg   | 0.00 mg  | 2.58 mg |

**Last comments on the actions** 

| Date                  | Comment                 |
|-----------------------|-------------------------|
| 17-04-2012            | OK Cancelled Prednisone |
| 16-04-2012            | OK Prednisone           |
| Siguiente diapositiva |                         |

← → ❤️ 📲 16 of 22 → 