



Universidad de Granada

[decsai.ugr.es](http://decsai.ugr.es)

## **Tema 2: ACM y Planificación Automática**

**Curso en Minería de Procesos (Gestión de Procesos)**

**Juan Fernández Olivares**



**DECSAI**

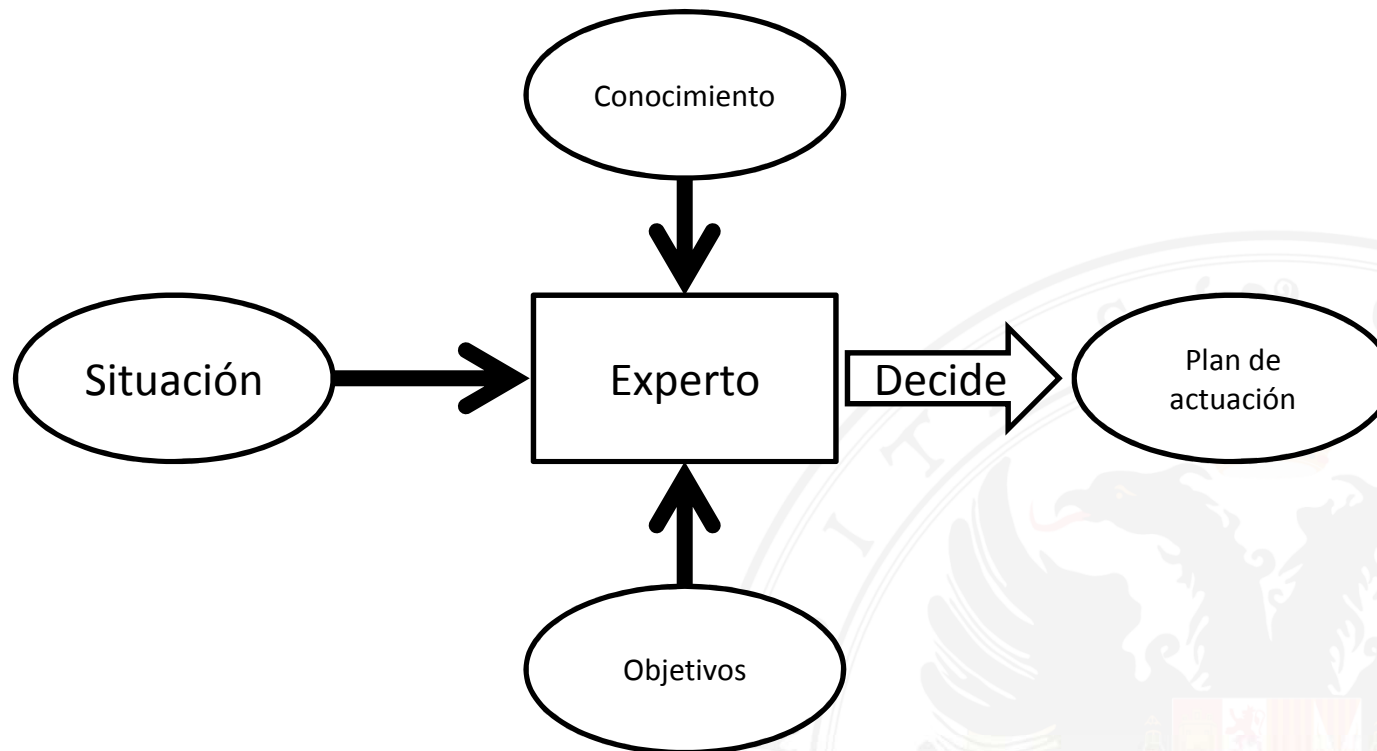
**Departamento de Ciencias de la  
Computación e Inteligencia Artificial**

En la sesión anterior ...



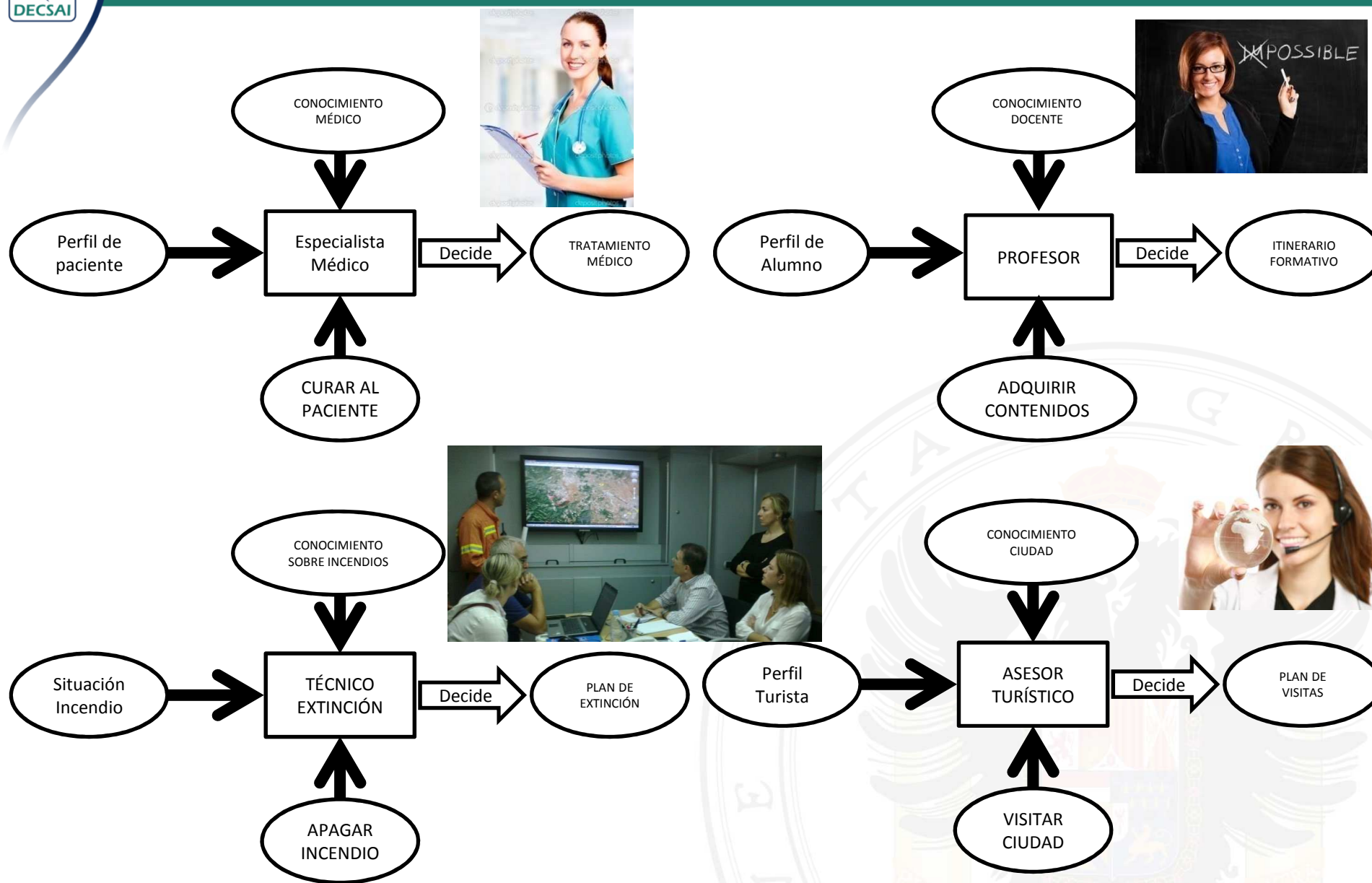
1. **Gestión Inteligente de procesos = Gestión de procesos del trabajo del conocimiento**
2. **Limitaciones BPM clásico para trabajo del conocimiento**
3. **Características trabajo del conocimiento**
4. **Adaptive Case Management (ACM)**
5. **Planificación automática como tecnología habilitadora de ACM**
  1. Planificación clásica
  2. Planificación HTN: Hierarchical Tasks Networks
  3. Modelos HTN extendidos
6. **Cognocare: aplicación de HTN para ACM**
7. **Monitorización y ejecución de planes.**

En esta sesión nos vamos a centrar en procesos que requieren conocimiento experto humano para su realización.



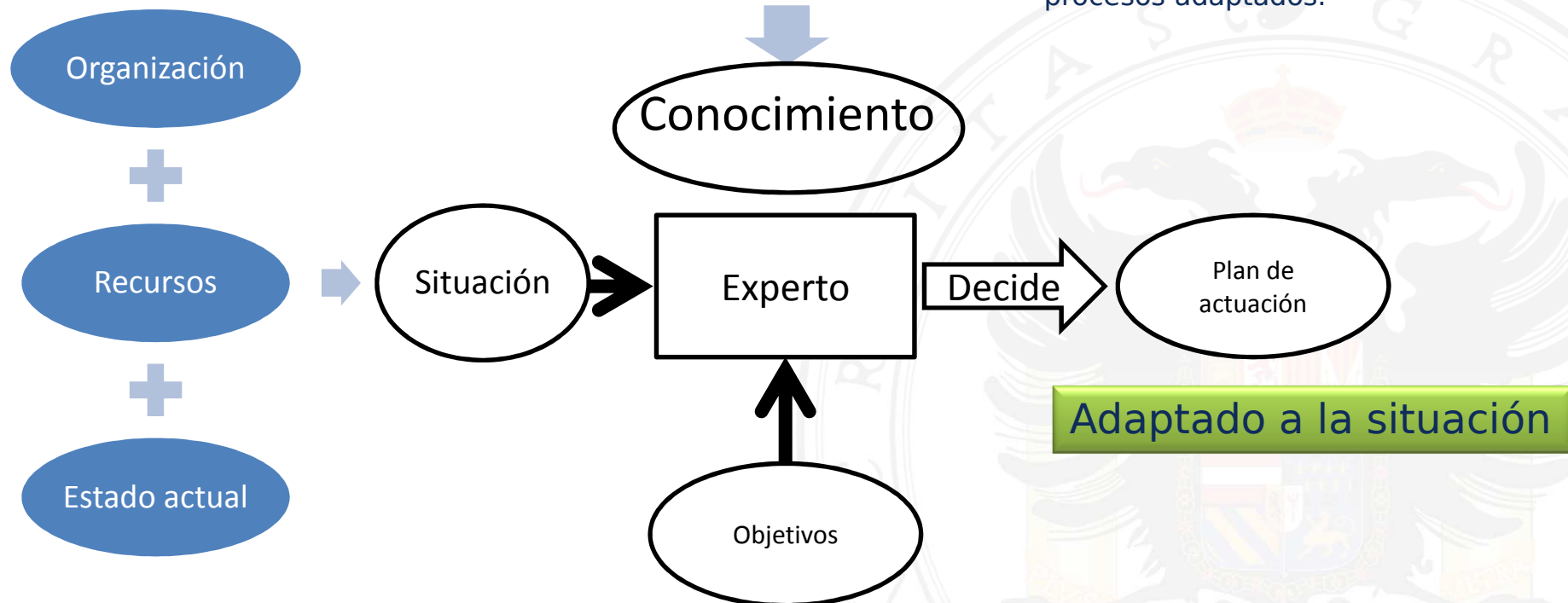
**DECISIÓN= SECUENCIA DE ACTIVIDADES**

**SOPORTE A LA DECISIÓN= RECOMENDACIÓN DE ACTIVIDADES**



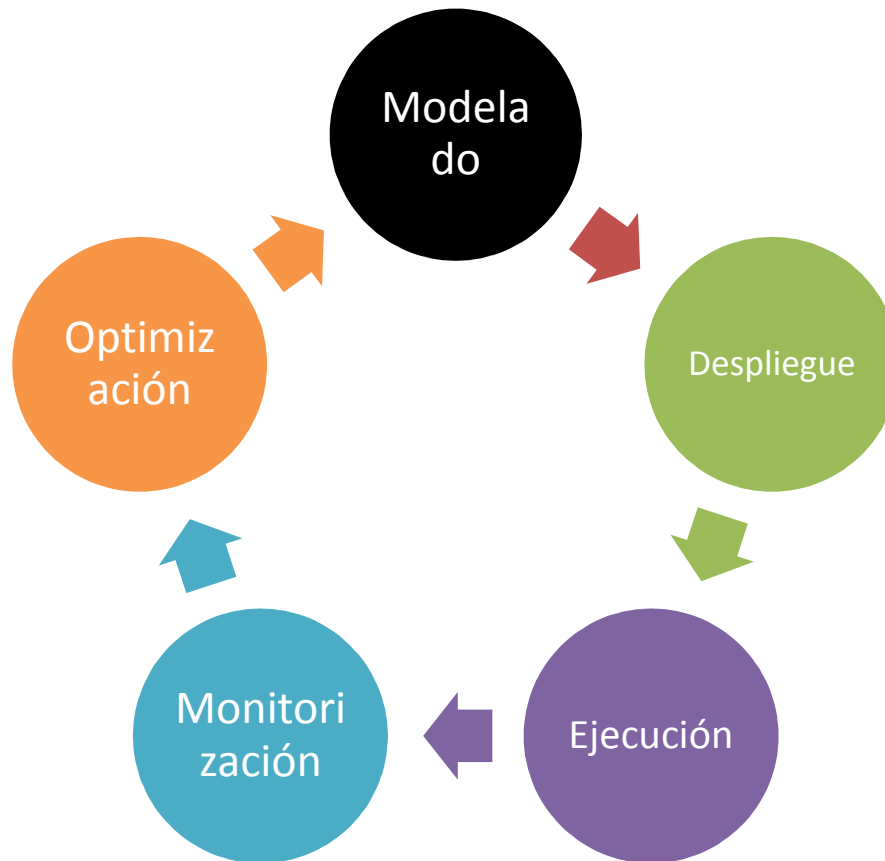
En estos ejemplos el experto utiliza su conocimiento junto a protocolos, normas o guías que debe respetar para elaborar un proceso o plan de actuación adaptado (tailored) a una situación concreta, para alcanzar un objetivo o desempeñar una tarea concreta.

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.

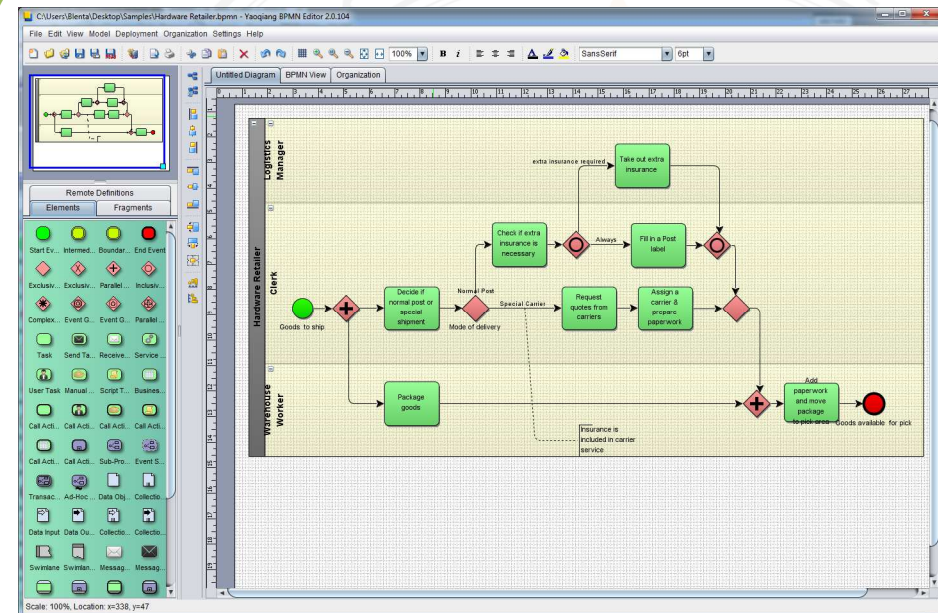




### SUITE BPM

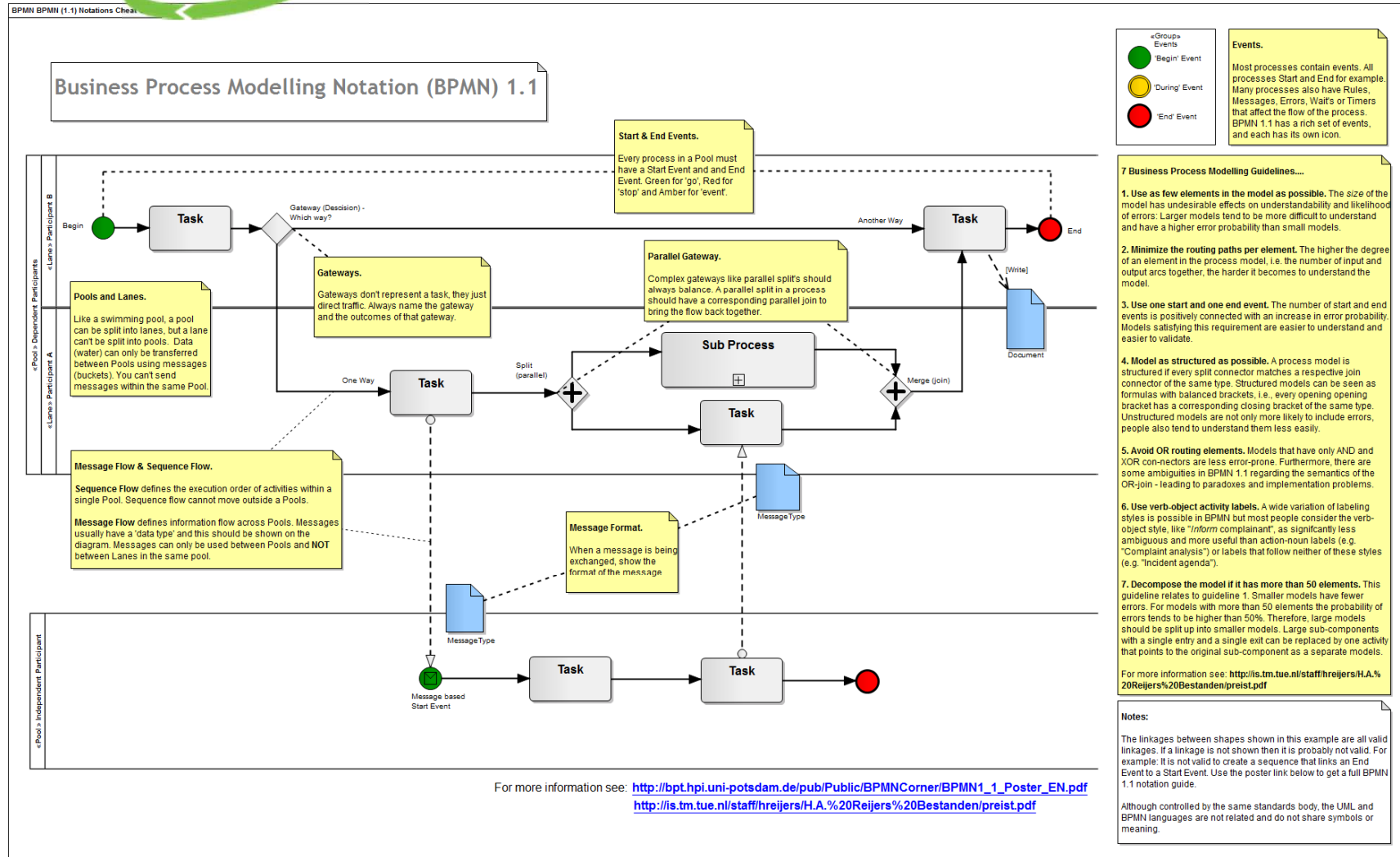


- Quién:
  - Analista de proceso.
- Qué:
  - Modelo de proceso.
- BPMN
  - Notación estándar para modelado.
  - <http://www.bpmn.org/>





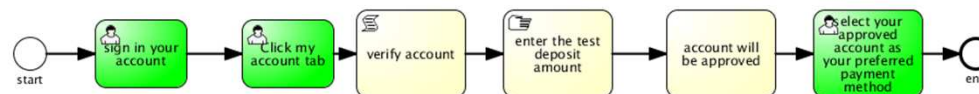
<http://www.bpmn.org>



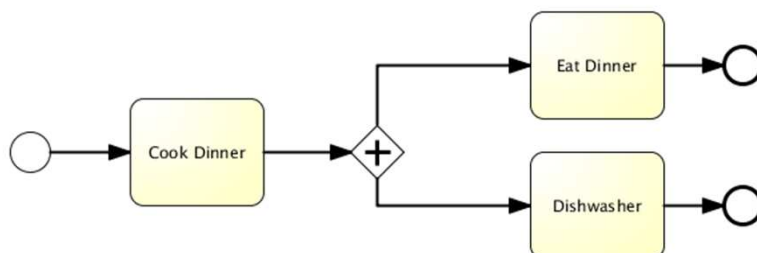


<http://bpmnhandbook.com/>

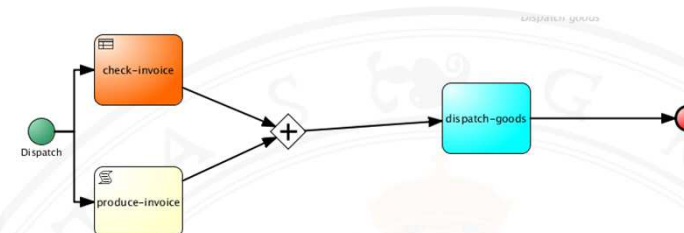
### Secuencia



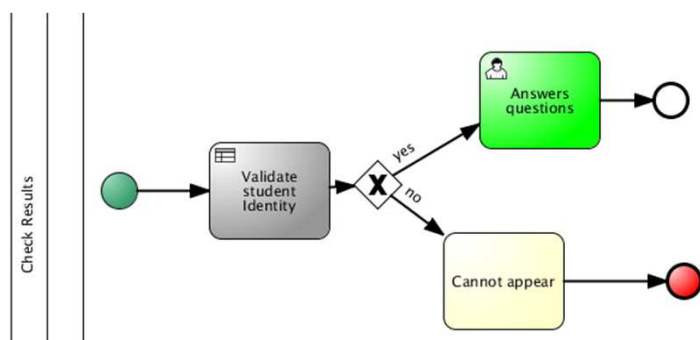
### Split paralelo



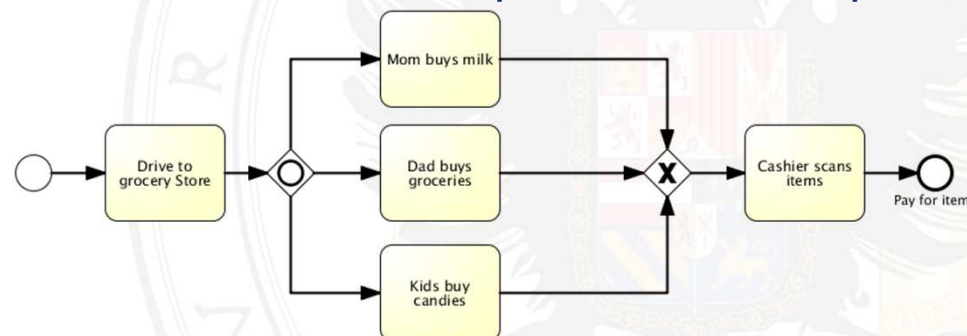
### Sincronización



### Selección exclusiva



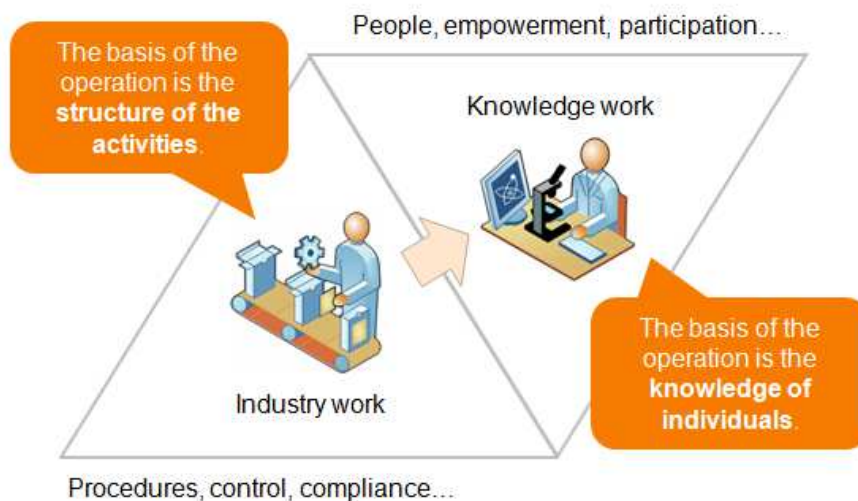
### Selección múltiple/Mezcla simple



Los **modelos de proceso usados en BPM**, 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 necesitan modelar "trabajo del conocimiento" y dar soporte a las decisiones y tareas diarias del "trabajador del conocimiento" o knowledge worker.



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.

- 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
    - ....

La información sobre la situación cambia constantemente, y 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.



Los modelos de proceso **detallados hasta el extremo**, llevan a instancias de procesos extremadamente rígidos. Ahora se exige que el plan pueda modificarse, pero manteniendo los objetivos para los que se ha planteado el proceso.



Rara vez ocurre que el trabajo de un experto se repita de la misma forma y varias veces para un mismo caso.

El curso de acciones depende de la situación, y la situación puede cambiar mucho, y varias veces, antes de que el trabajo haya finalizado. Y por tanto la secuencia de acciones ante cada situación no se puede conocer por adelantado

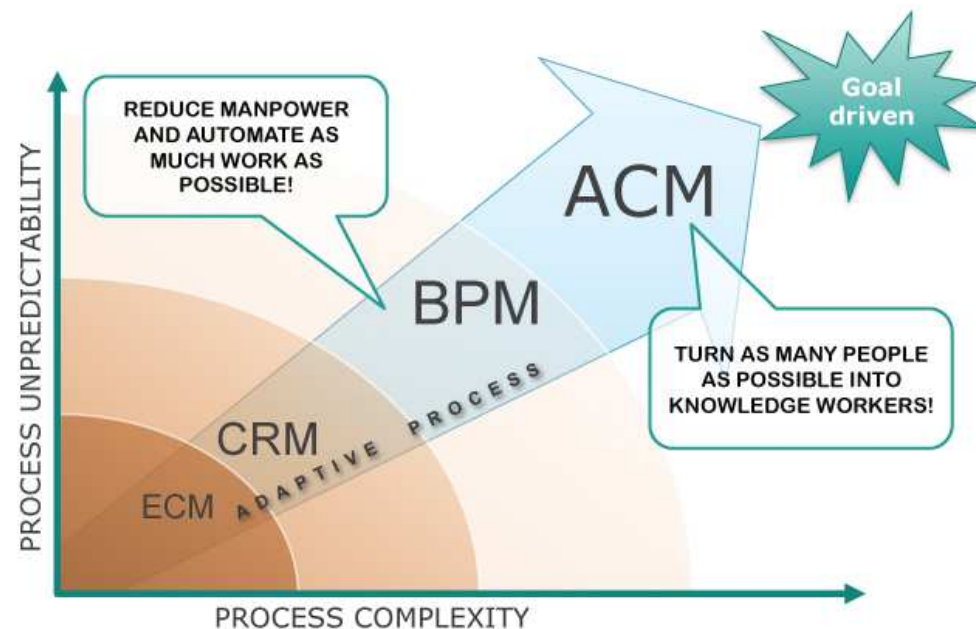
Un proceso de trabajo del conocimiento se despliega a medida que avanza. Una etapa previa ofrece nuevo conocimiento y este nuevo conocimiento determina la nueva etapa a realizar.



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.

- Preguntas:
  - ¿Cómo podemos modelar procesos orientados a objetivos que incorporen conocimiento experto?
  - ¿Cómo se pueden sintetizar 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 de procesos con estas características? (siguiente sesión)



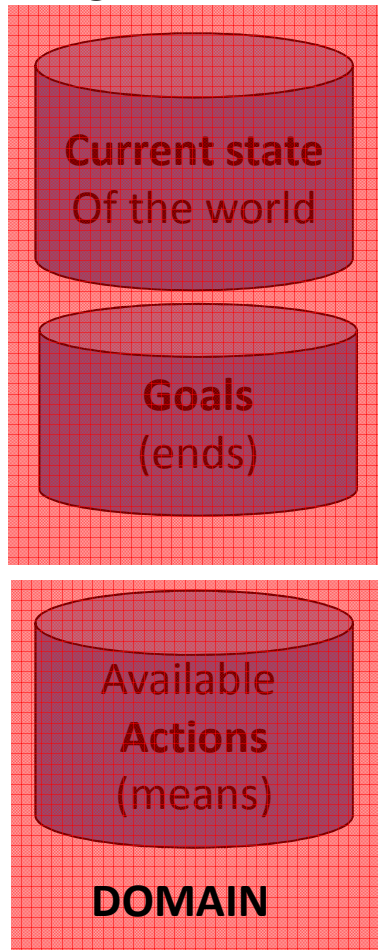
- Respuestas:
  - Vienen de la mano de la **Planificación Automática**.



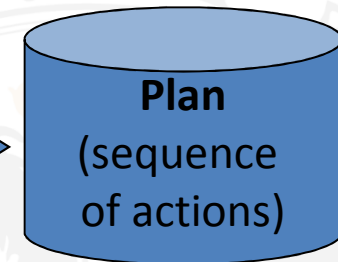
Nau, Dana, Malik Ghallab, and Paolo Traverso. *Automated Planning: Theory & Practice: Theory and Practice*. Amsterdam ; Boston: Morgan Kaufman Publ Inc, 2004. **(2nd Edition 2016)**

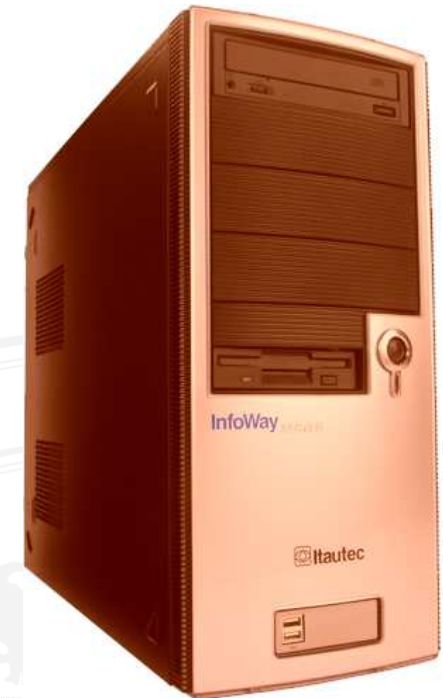
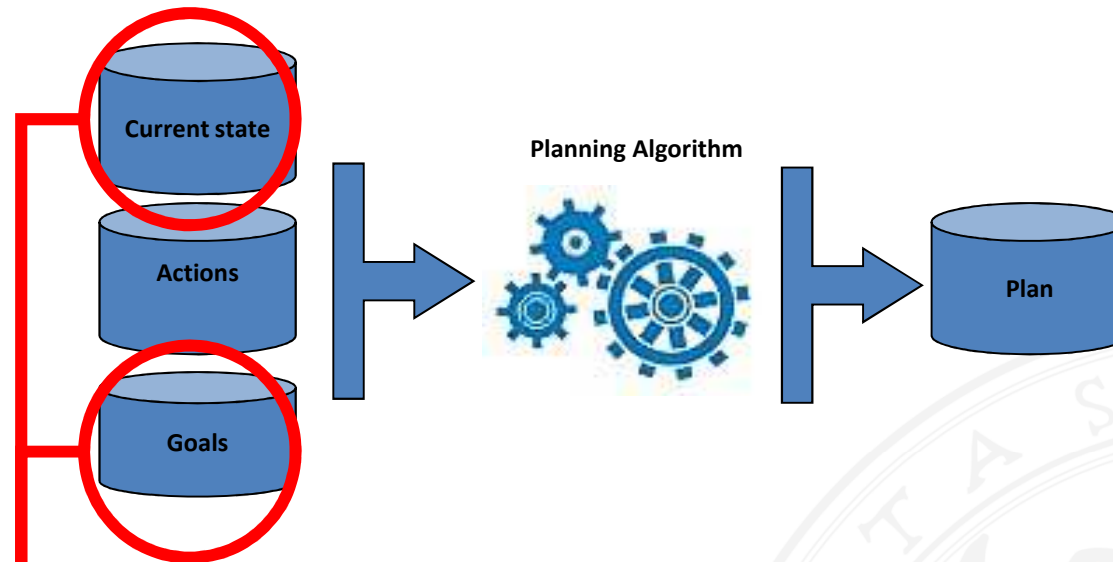
La **Planificación Automática** (Automated Planning) es una disciplina que estudia la **generación de planes** (procesos) a partir de un **modelo de acciones**, de unos **objetivos** y una **situación** de partida.

### PROBLEM



### Planning Algorithm

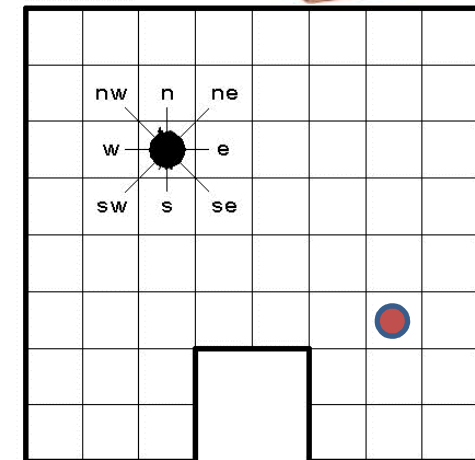




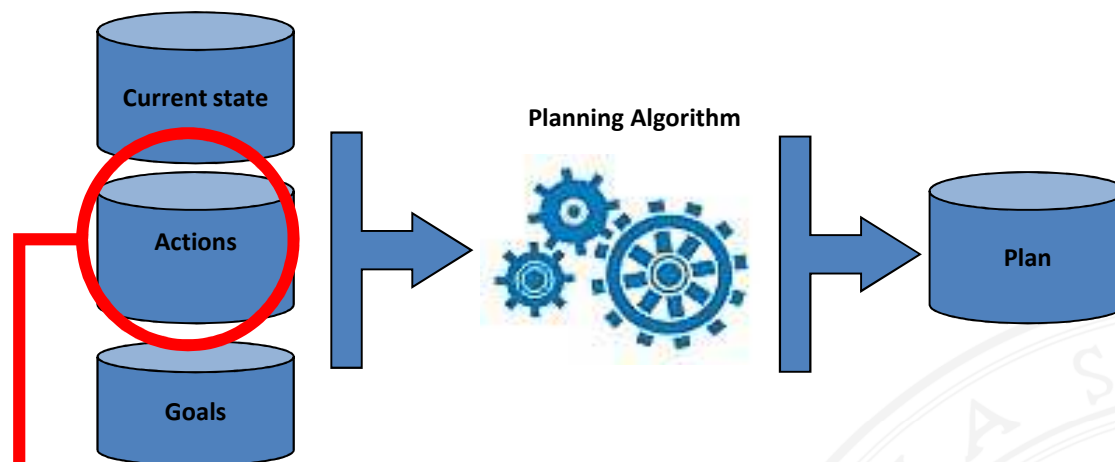
### I. Formal representation languages

- 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



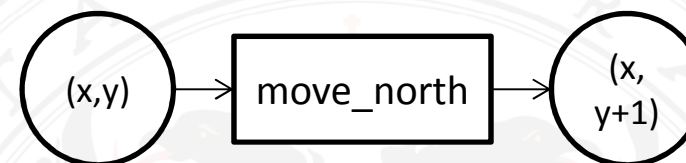
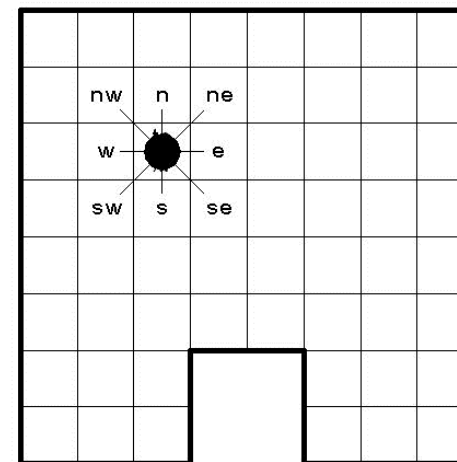
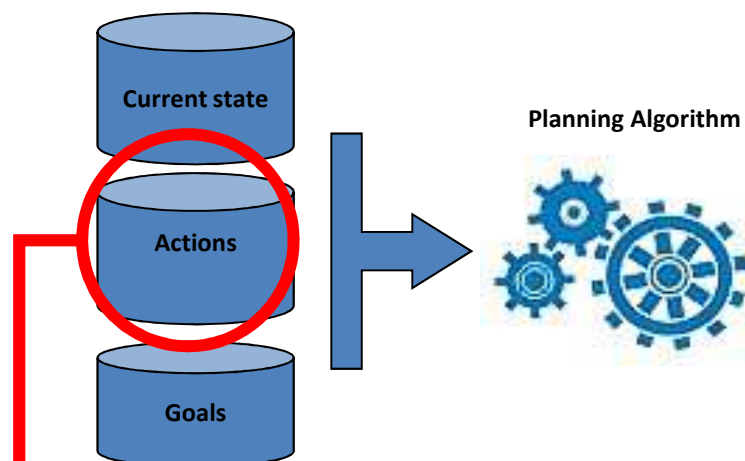
#### II. Models of action: Representing change in the environment

- Predicate logic
- Structured models

DOMAIN





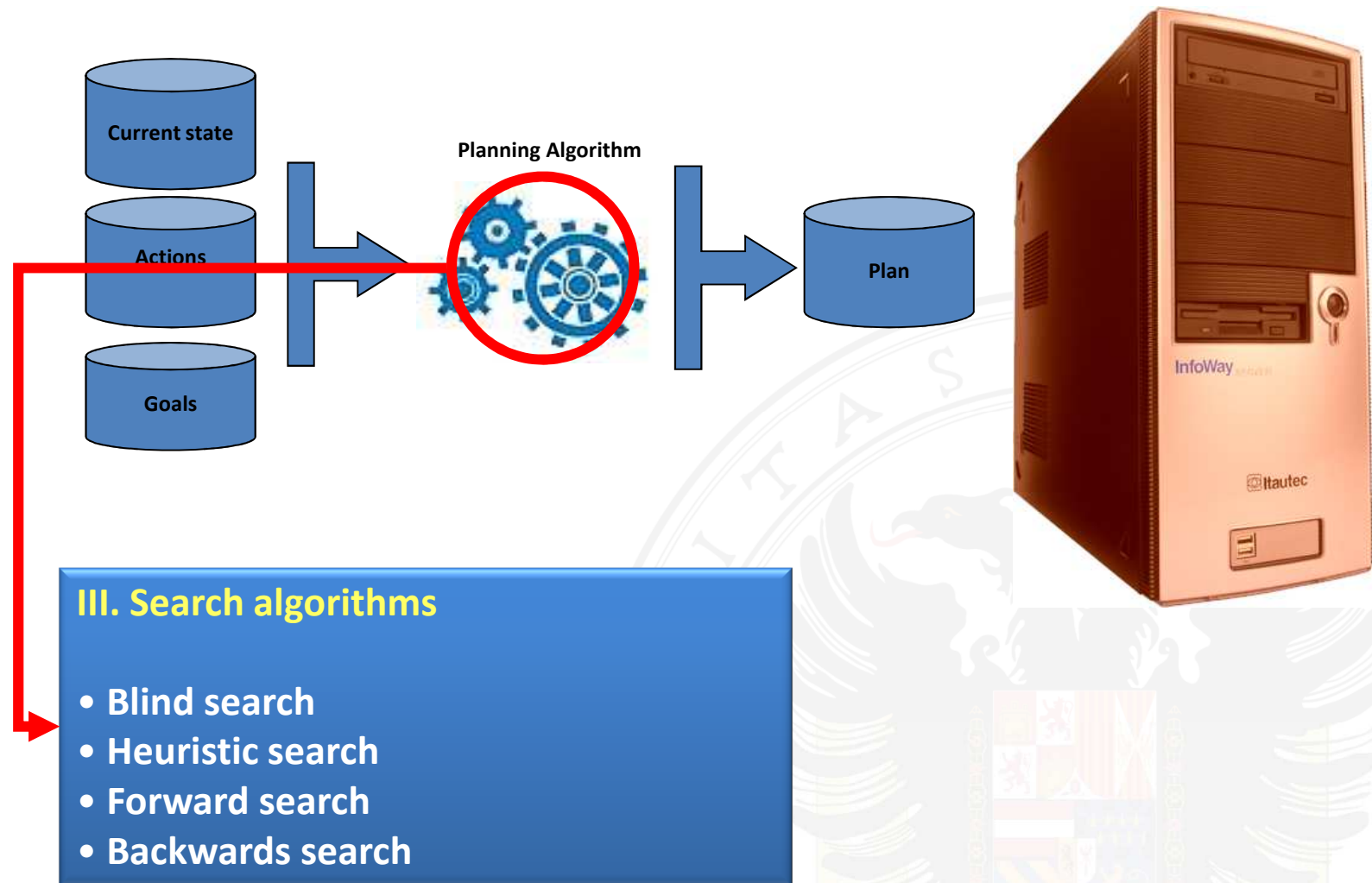


## II. Models of action: Representing change in the environment

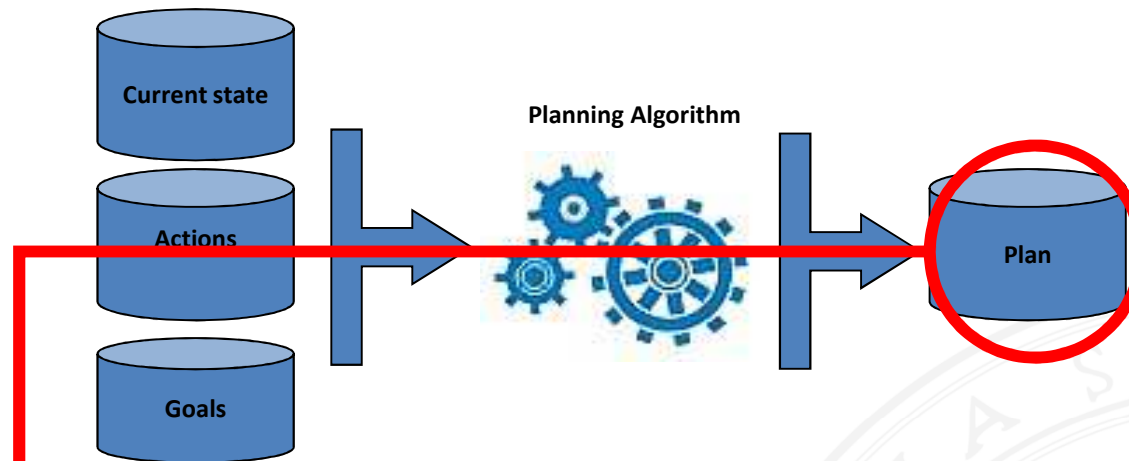
- Predicate logic
- Structured models

DOMAIN

```
action: move_north(robot)
Prec: at(robot, x,y)
Effect: at(robot, x, y+1)
```

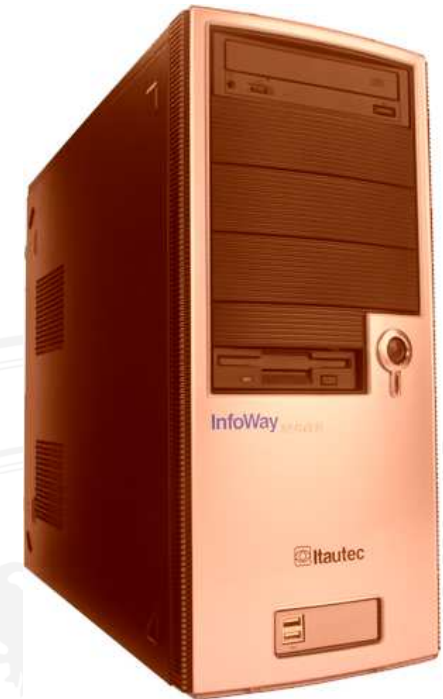


### Artificial Intelligence planning



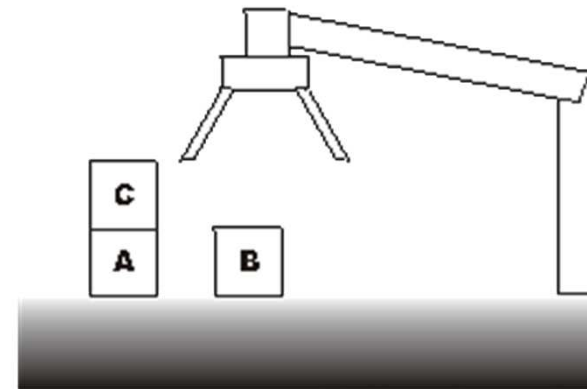
#### IV. Sequences of actions:

- Totally ordered
- Partially ordered
  - Independence
  - Parallelism



- Planificación automática:
  - resolución de problemas cuya solución es una secuencia de actividades.
  - que deben ejecutarse para alcanzar un objetivo desde una situación de partida.
- Visión 1: Planificación automática como **Resolución de problemas**:
  - Estado inicial
  - Estado objetivo
  - Operadores (Dominio de planificación)
  - Proceso de búsqueda
  - Solución: secuencia de actividades -> plan
  - Genera una secuencia de actividades, pero no una cualquiera, sino una que define el comportamiento de un agente autónomo para alcanzar un objetivo.
- Visión 2: Planificación automática como Gestión de procesos
  - Un **planificador es un Generador de procesos**.
  - Secuencia de actividades, un plan, es un proceso.
  - Entonces 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

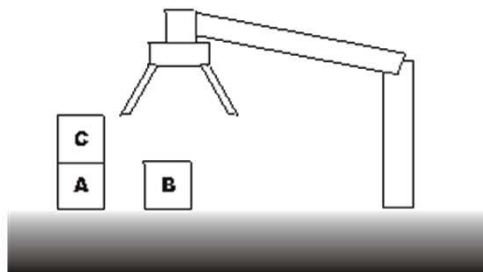
Pickup(X)

Putdown(X)

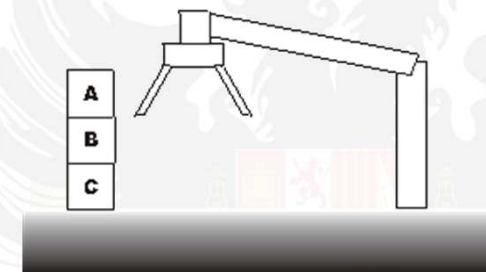
Stack(X,Y)

Unstack(X,Y)

#### Estado Inicial



#### Goal

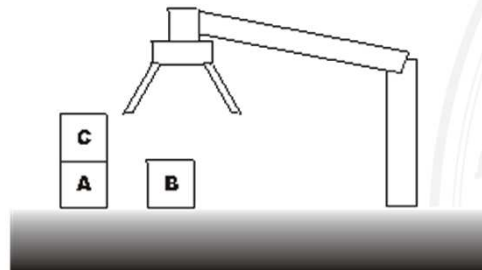


Pickup(C), Putdown(C), Pickup(B), Stack(B,C), Pickup(A), Stack(A,B)

#### 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

Initial state



→  $\neg \text{clear}(B)$

$\text{handempty}, \text{clear}(C), \text{on}(C,A), \text{ontable}(A),$   
 $\text{ontable}(B)$

$\text{clear}(B)$



- Representación 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



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)

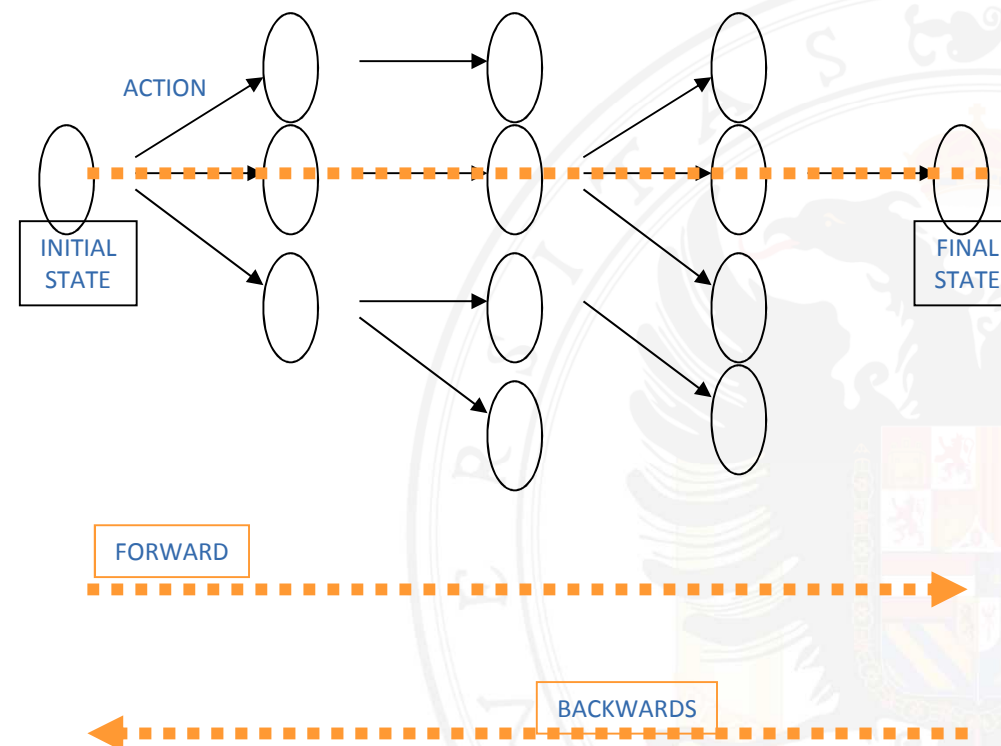
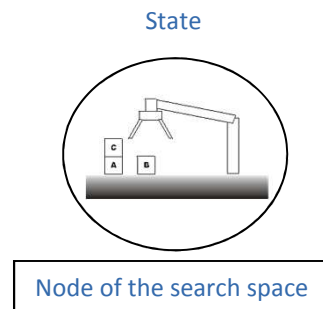
Delete: handempty, clear(X), on(X,Y)

Add: holding(X), clear(Y)

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

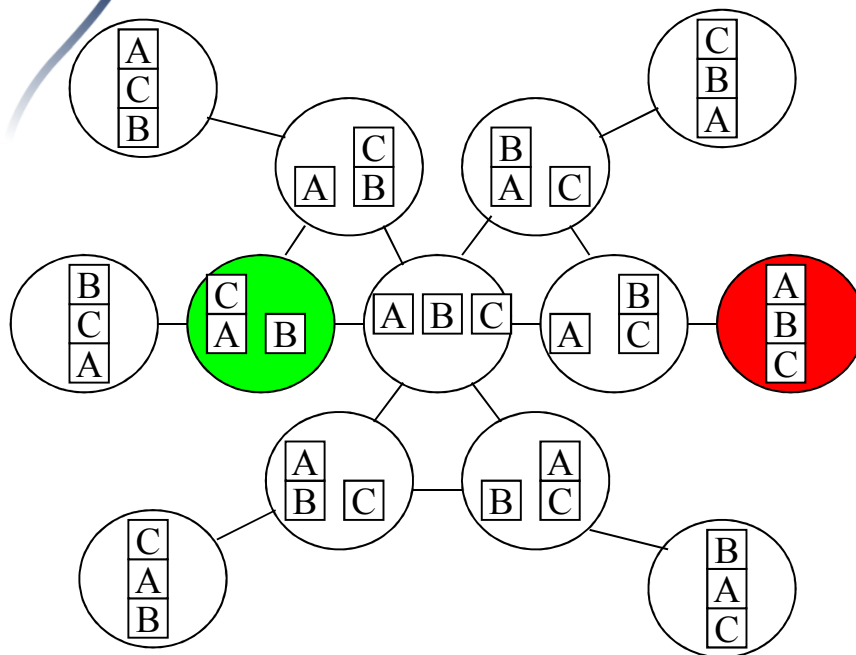
[https://en.wikipedia.org/wiki/Planning\\_Domain\\_Definition\\_Language](https://en.wikipedia.org/wiki/Planning_Domain_Definition_Language)

- 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.*

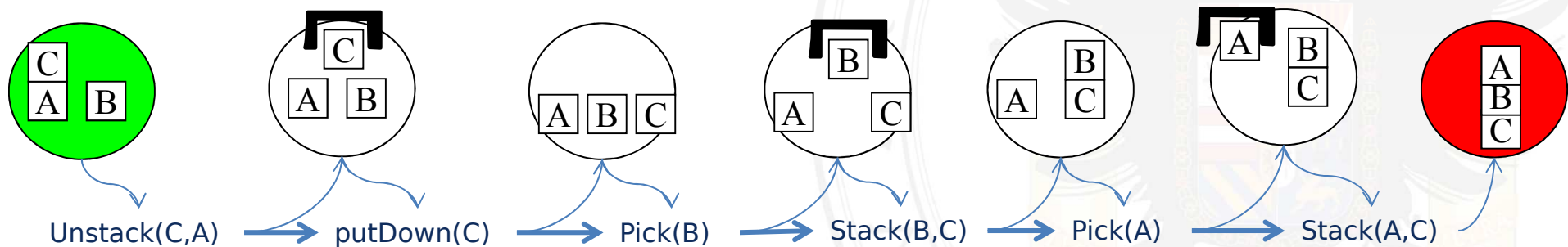


- 

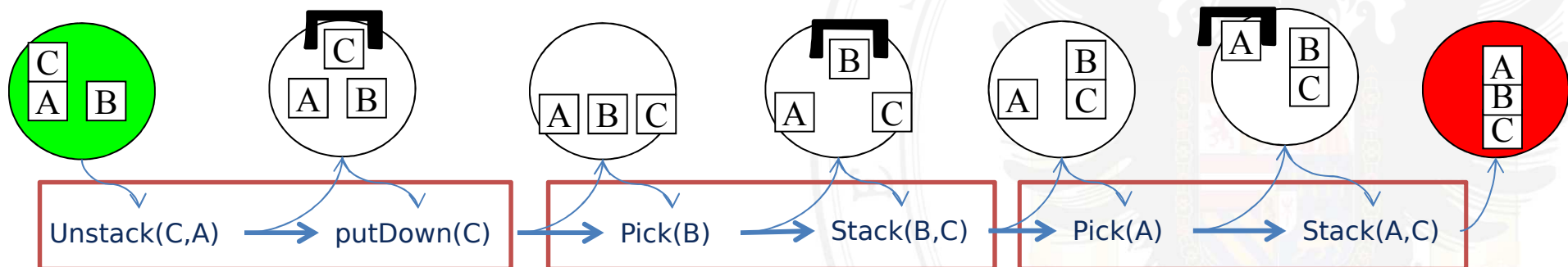
Helmert, Malte. "The Fast Downward Planning System." *J. Artif. Intell. Res. (JAIR)* 26 (2006): 191–246.



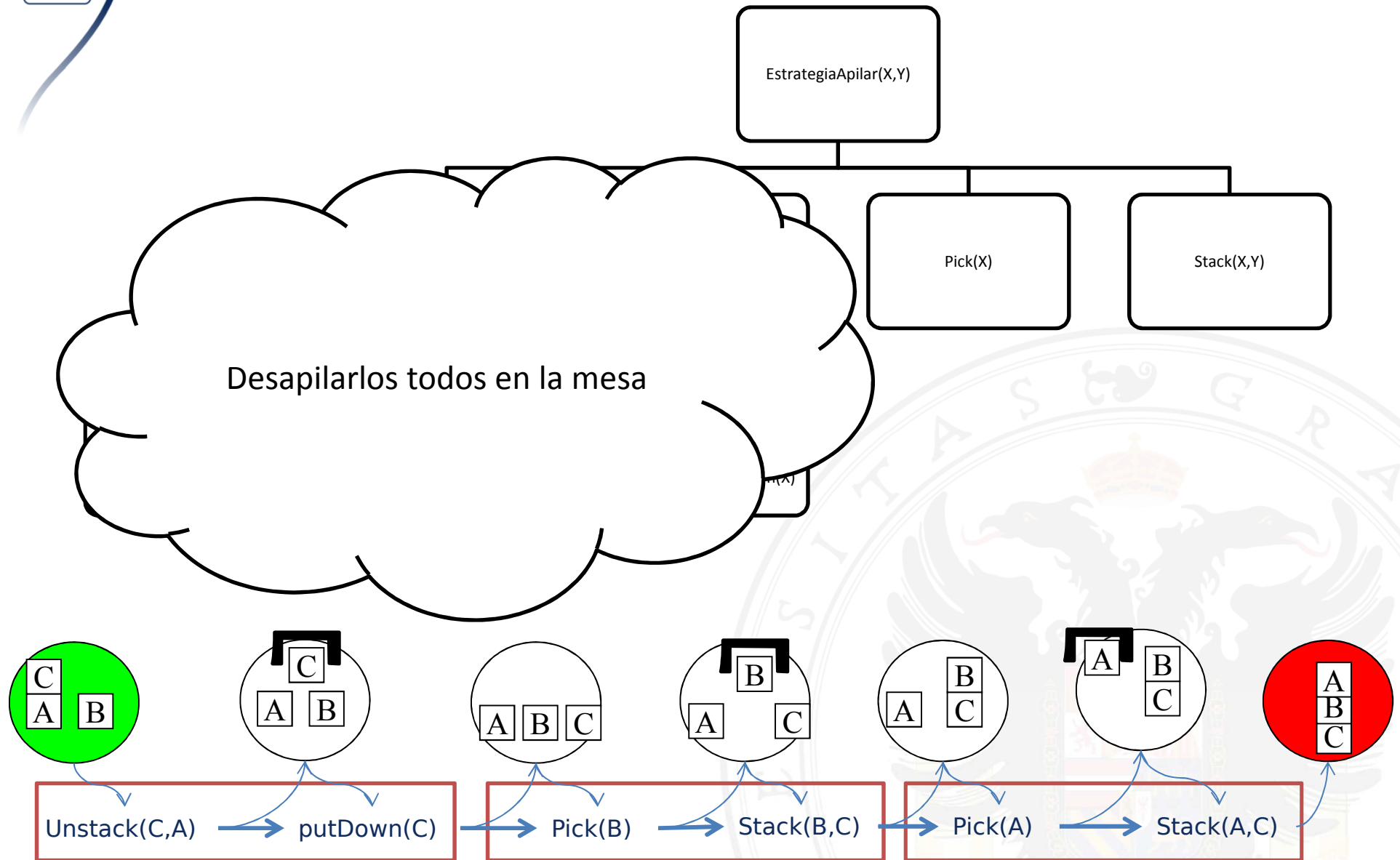
- El plan obtenido es una secuencia totalmente ordenada (en este ejemplo)
- El camino entre el estado inicial y el final.
- Secuencia de pasos para alcanzar el objetivo.
- Síntesis de un proceso orientado a objetivos.

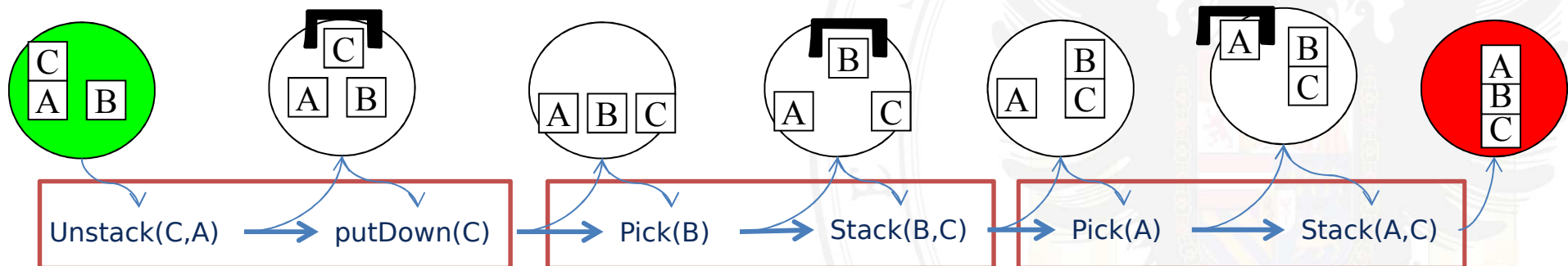
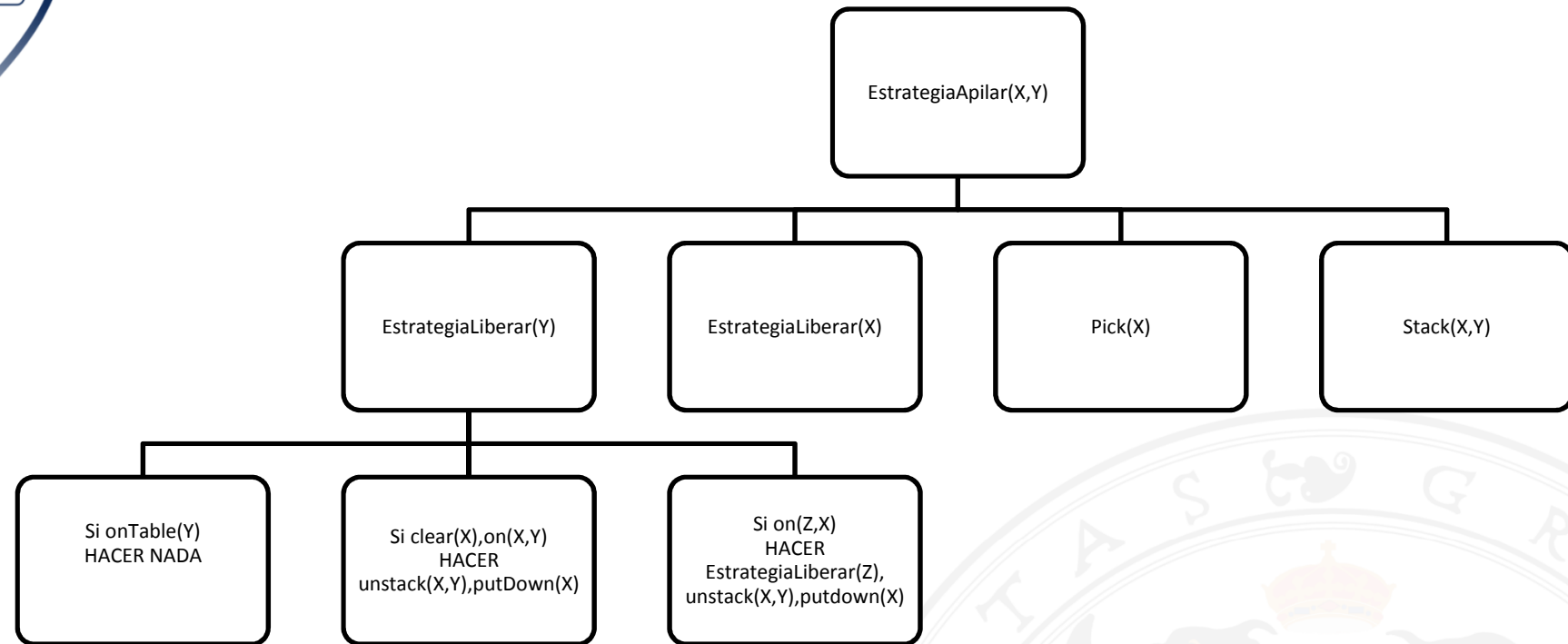


- Ventajas:
  - Resolución de problemas combinatorios.
- Limitaciones:
  - Pueden explorar caminos claramente desechables.
  - Centrado en analizar relaciones causa-efecto, dificultad en representar otras relaciones entre acciones, como patrones de proceso comunes, habituales en actividades humanas.
- ¿Podemos mejorar la forma en que se resuelven problemas de planificación, tratando de reflejar cómo lo hacemos los humanos?

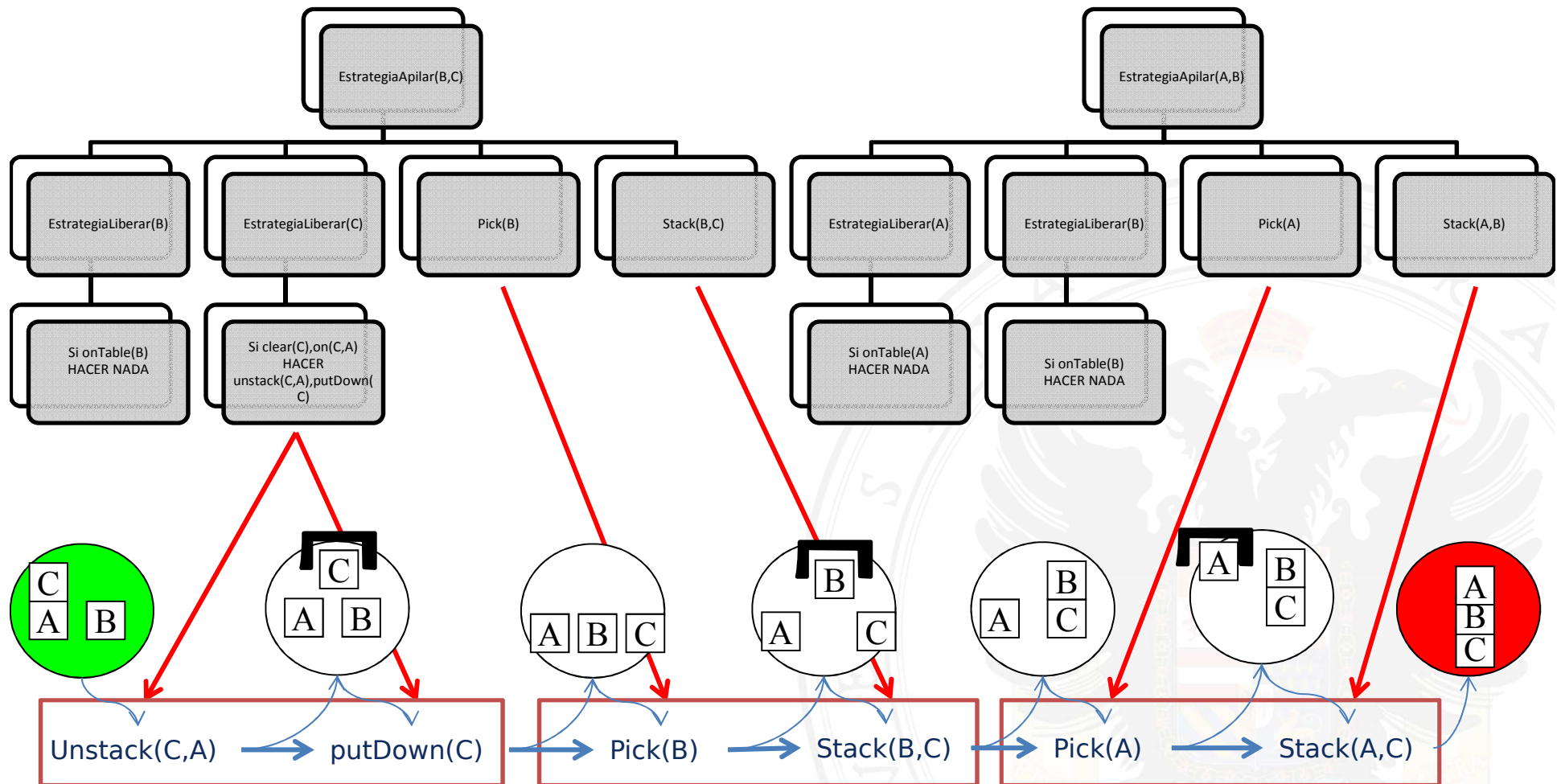




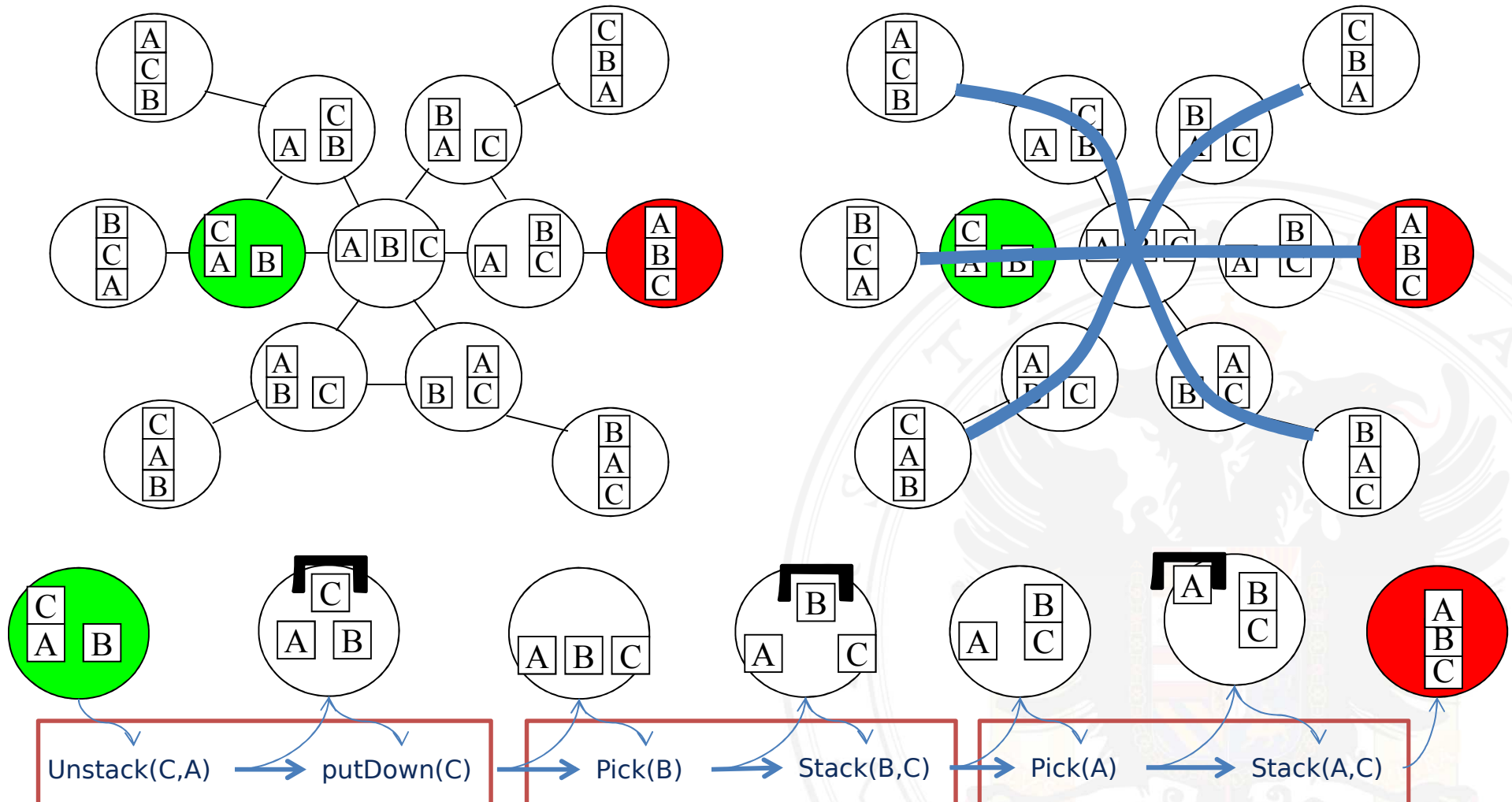




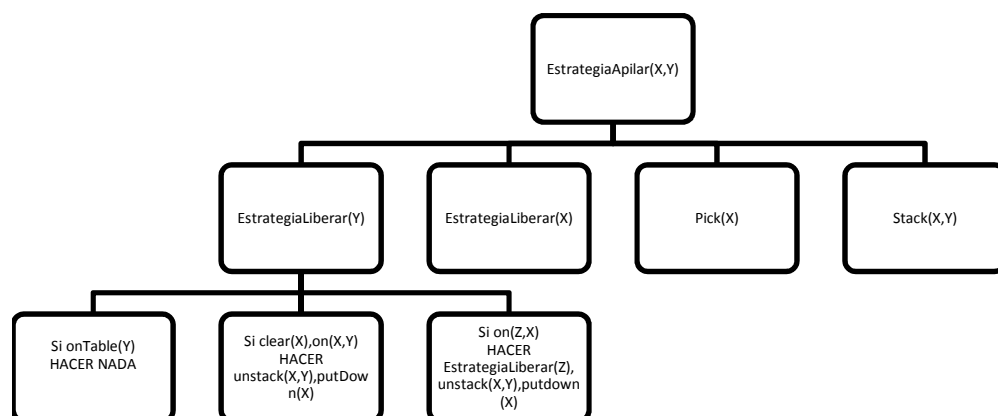
**Planificación jerárquica:** técnica de planificación que permite representar estrategias a distintos niveles de abstracción para resolver problemas de planificación.



- 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.



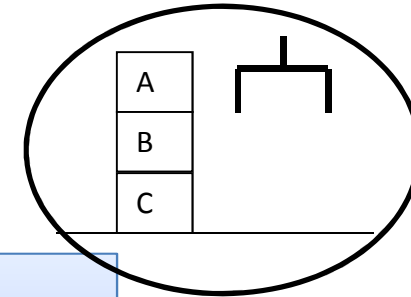


- Planificación HTN (Hierarchical Task Networks): una técnica de planificación jerárquica en la que el dominio se representa a partir de:
  - Un conjunto de **tareas compuestas**
  - Para las que se especifican alternativas de descomposición mediante **métodos de descomposición**
  - Un conjunto de **tareas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

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



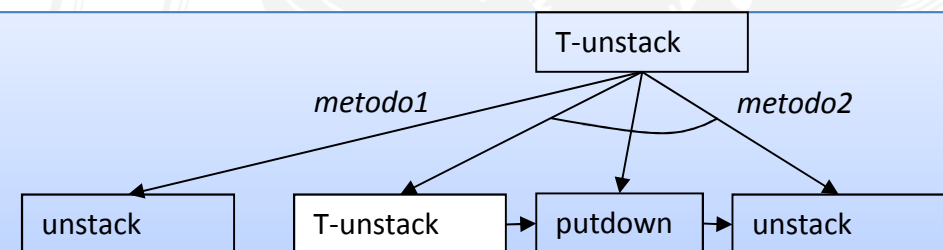
```
(:action unstack
:parameters (?x ?y - block)
:condition (and (handempty) (clear ?x))
:effect (and (not (handempty)) (holding ?x) (clear ?y)
(not (clear ?x))))
```

- 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))))
```

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

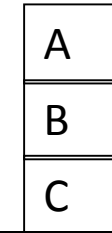
(:action putdown
:parameters (?x - block)
:precondition (holding ?x)
:effect (and (ontable ?x) (clear ?x) (handempty) (not (holding ?x))))
```

```
(: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

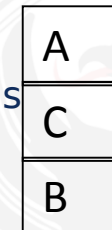
(on A B) (on B C) (ontable C)  
(clear A) (handempty)



- Objetivo

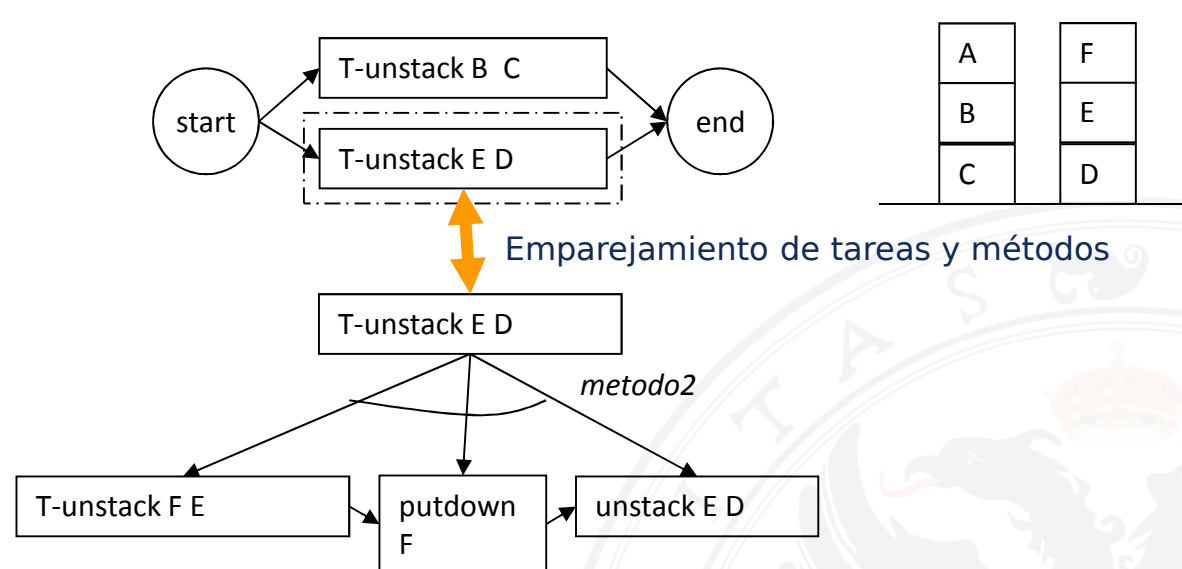
- Clásico (No HTN): (on C B)(on A C)
  - Interpretación: se desea alcanzar dos objetivos
  - No se pueden** especificar estrategias

- 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



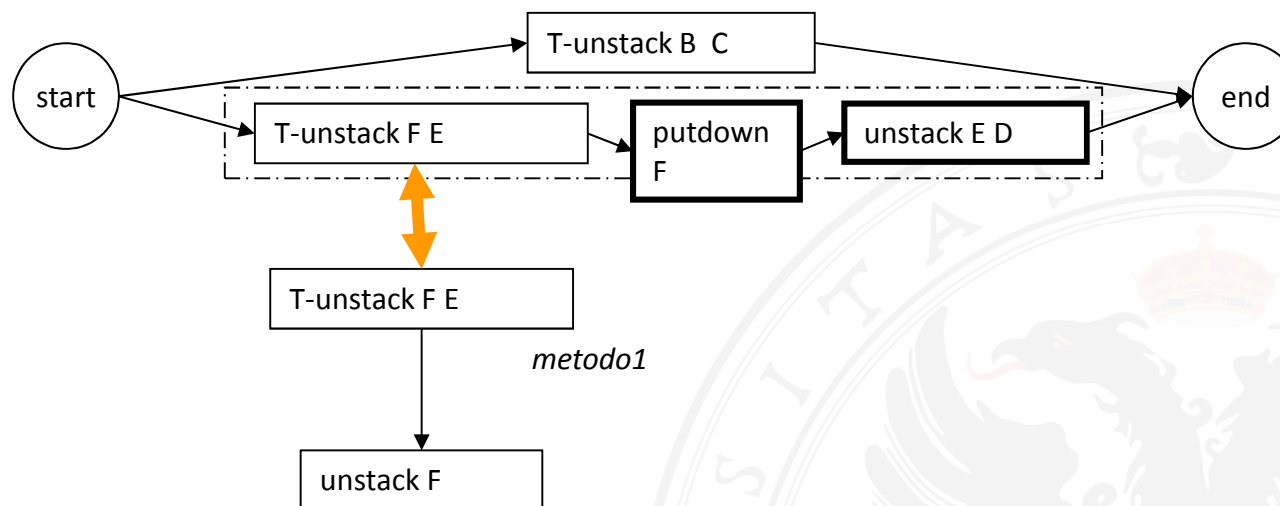
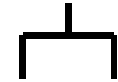
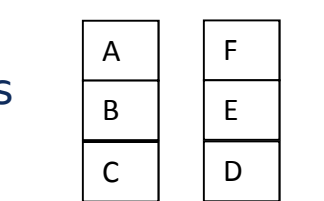
### Objetivo de planificación

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



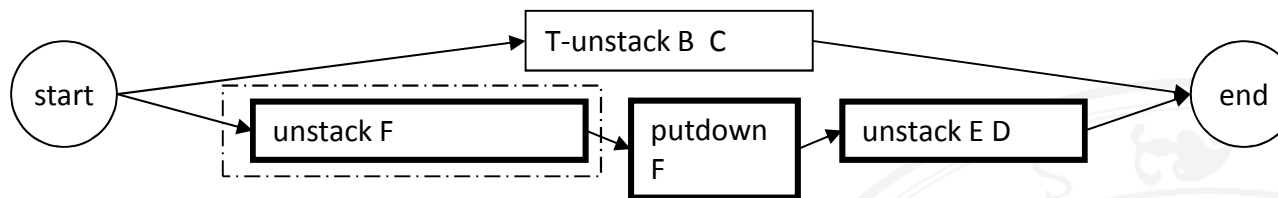
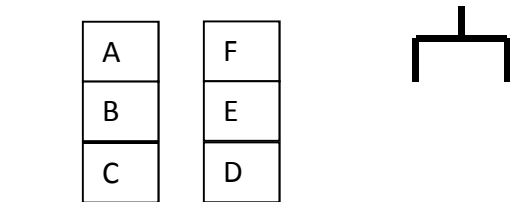
- Proceso de planificación: reducir (descomponer) tareas compuestas aplicando métodos de descomposición.

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

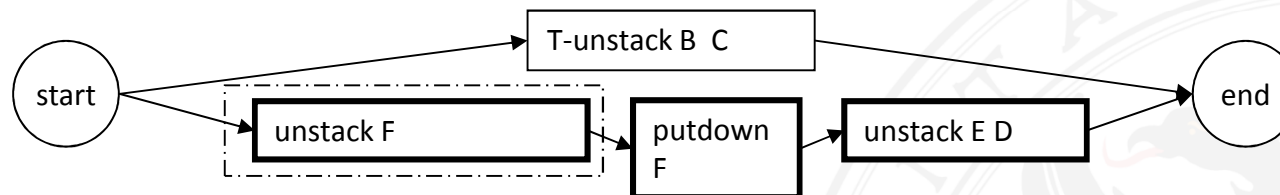




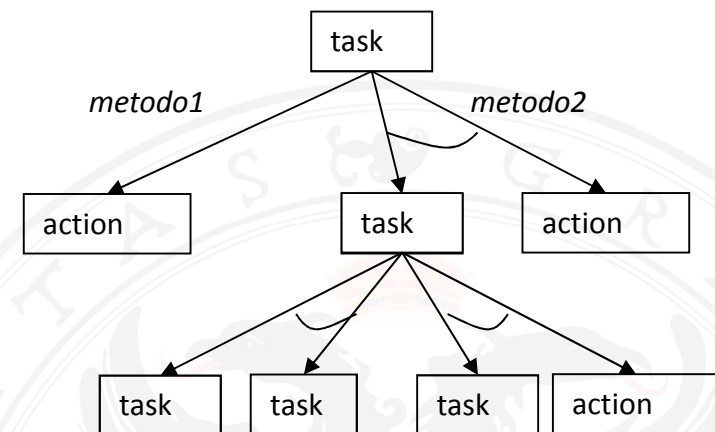
- Planning goal
  - El proceso de reducción de una tarea finaliza cuando todas las subtareas son primitivas.



- 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.

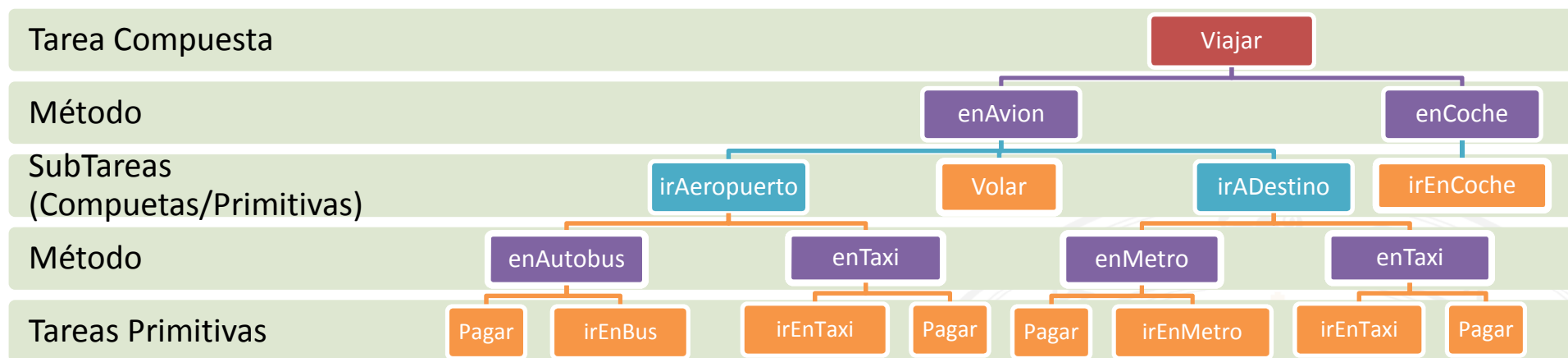


- Los operadores de reducción dan lugar a un espacio de búsqueda diferente al de planificación clásica:
  - Nodos: tareas/acciones
  - Arcos: conectan nodos relacionados con métodos de reducción
- Búsqueda en un grafo Y/O.



- Veamos un ejemplo de búsqueda primero en profundidad.

### 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)

(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

Pagar

irEnBus



### 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 refinar más.*
    - *devuelve el plan completamente refinado*
- Plan: el camino desde el estado inicial hasta el objetivo.

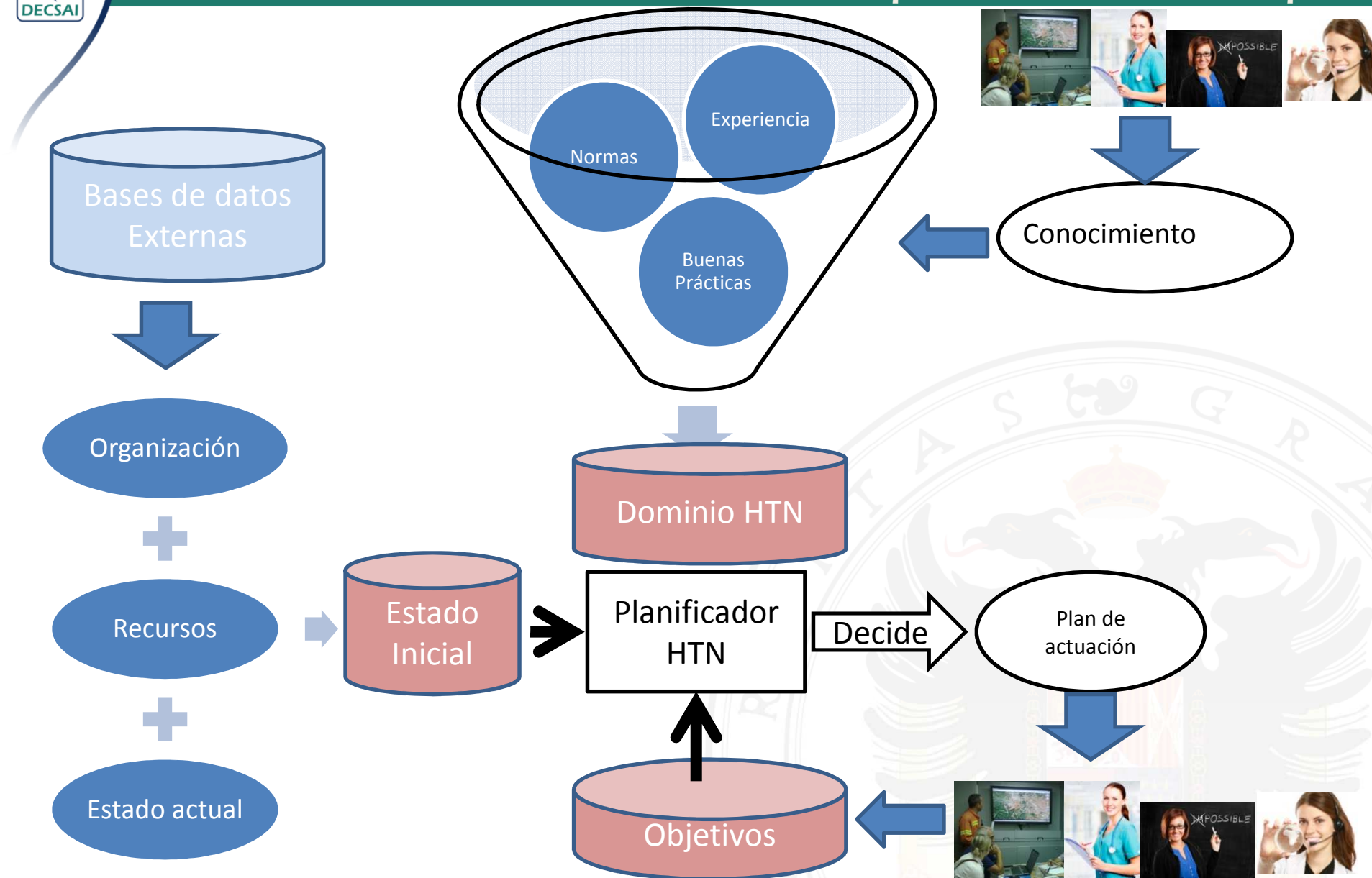
### Planificación jerárquica: resolución de problemas usando conocimiento experto.

- 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:
  - acciones equivalente a primitivas
  - 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 árbol (o bosque) de descomposición hasta encontrar una red de tareas sólo primitivas.
- Plan: una red de tareas primitivas.

• 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.





- 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 patterns 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.

- [illegible]

F. Palao, J. Fdez-Olivares, L. Castillo, y O. Garcia, «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.

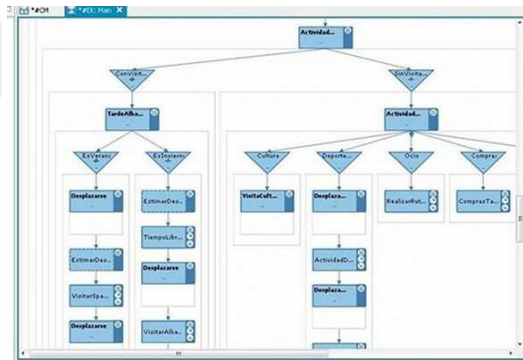




### Dominio



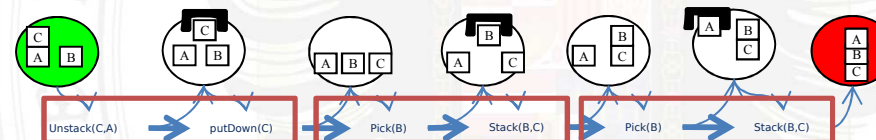
### Objetivos



Ingeniero  
Conocimiento

### Contexto

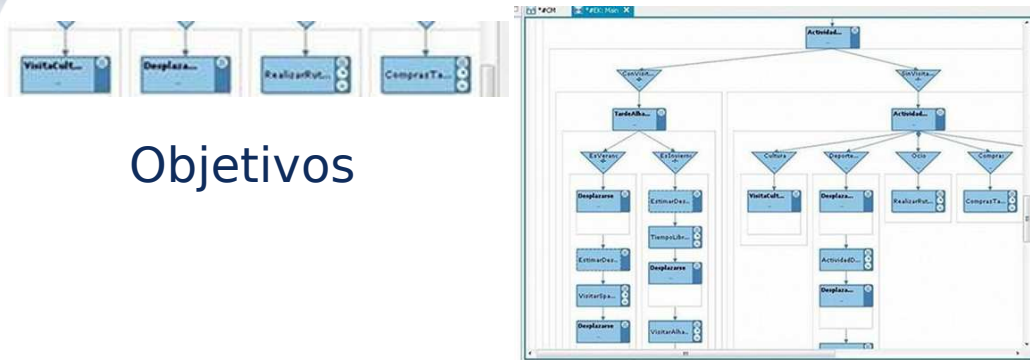
Person	Address	City	Country	Estimate	Lat
1	1000 East Street	Phoenix City	London	London	Don
2	154 Bally Vink	154 Bally Vink	Exonator	Exonator	Exon
3	802 Tennessee Avenue	Exonator	Tennessee	Exonator	Exon
4	700 Fourth Lane	Tennessee	Tennessee	Exonator	Exon
5	1070 Beach Way	Exonator	Exonator	Exonator	Exon
6	700 North Lane	Exonator	Exonator	Exonator	Exon
7	910 Eighth Way	Exonator	Exonator	Exonator	Exon
8	247 Fifth Place	Exonator	Exonator	Exonator	Exon
9	840 State Street	Exonator	Exonator	Exonator	Exon
10	740 Washington Street	Exonator	Exonator	Exonator	Exon
11	300 Tennessee Place	Exonator	Exonator	Exonator	Exon
12	14 Central Place	Exonator	Exonator	Exonator	Exon
13	600 Lower Avenue	Exonator	Exonator	Exonator	Exon
14	800 Third Street	Exonator	Exonator	Exonator	Exon
15	220 Broadway Place	Exonator	Exonator	Exonator	Exon
16	1000 Garden Place	Exonator	Exonator	Exonator	Exon







### Dominio



Ingeniero  
Conocimiento

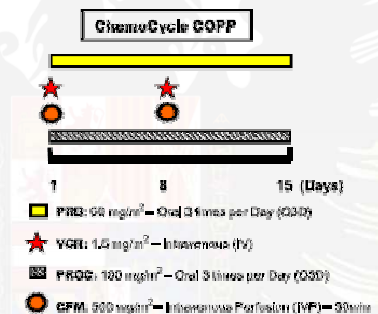
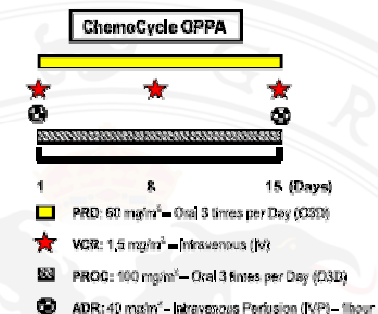
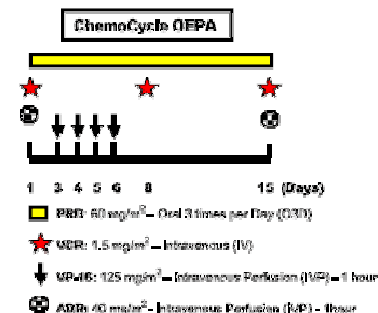
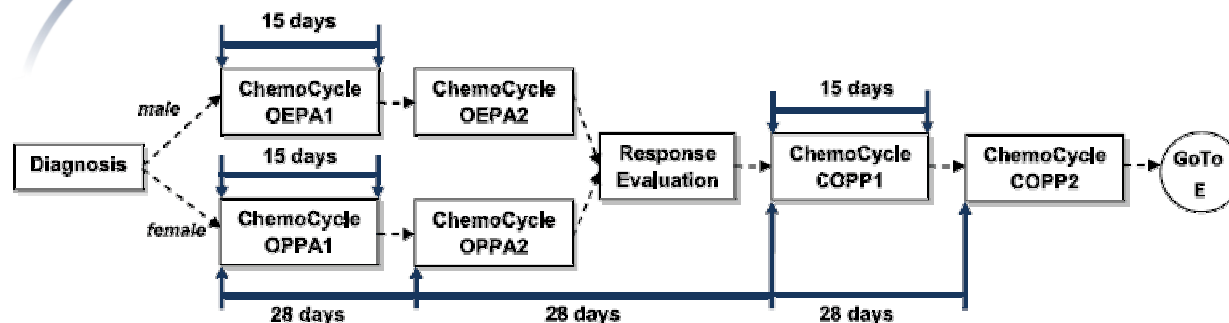
### Contexto

Person	Address	City	Postcode	Country	Age
1	1000 East Street	Phoenix City	Phoenix City	London	25
2	154 Bally Vink	154 Bally Vink	Exonisor	Exonisor	25
3	802 Tennessee Avenue	502 Tennessee Avenue	Exonisor	Exonisor	25
4	702 Fourth Lane	702 Fourth Lane	Tenderbark	Tenderbark	25
5	1078 Beach Way	1078 Beach Way	Com Midlow	Com Midlow	25
6	700 North Lane	700 North Lane	North Beach	North Beach	25
7	910 Eighth Way	722 Arroyo Lane	Madison Park	Columbus	25
8	247 Fifth Place	247 Fifth Place	Western Addition	Western Addition	25
9	843 Station Street	843 Station Street	High Valley	High Valley	25
10	740 Washington Street	740 Washington Street	Civic Center	Civic Center	25
11	300 Tennessee Place	300 Tennessee Place	Polkman's Village	Polkman's Village	25
12	14 Central Place	14 Central Place	Buena Vista	Buena Vista	25
13	600 Lower Avenue	600 Lower Avenue	Diamond Heights	Diamond Heights	25
14	800 Third Street	800 Third Street	Civic Center	Civic Center	25
15	220 Broadway Place	220 Broadway Place	Olden View	Olden View	25
16	1018 Market Gardens Place	1018 Market Gardens Place	Polkman Hill	Polkman Hill	25



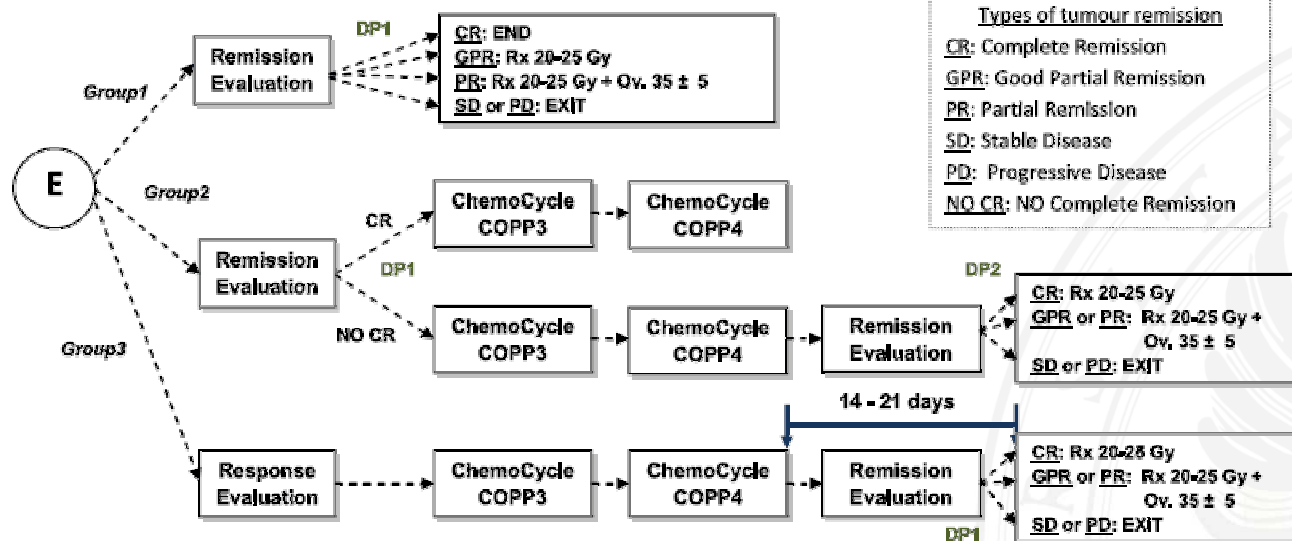
- Ejemplo HPDL usado para
  - representar un protocolo clínico en oncología pediátrica.
  - representar estado del paciente, recursos del hospital
  - representar las tareas objetivo a cumplir para tratar al paciente.





**Types of tumour remission**

CR: Complete Remission  
GPR: Good Partial Remission  
PR: Partial Remission  
SD: Stable Disease  
PD: Progressive Disease  
NO CR: NO Complete Remission



**Evaluation tests**

physical tests, complete blood count (CBC), renal function, liver function, erythrocyte sedimentation rate (ESR), Xray Thorax, abdominal ultrasound, adenopathy ultrasound

**DP1**: First decision point  
**DP2**: Second decision point  
**Rx**: Radiotherapy (5 sessions / week)  
**END**: End of Treatment  
**EXIT**: The patient leaves the protocol since the health conditions are not favorable

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

Condiciona

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")
```

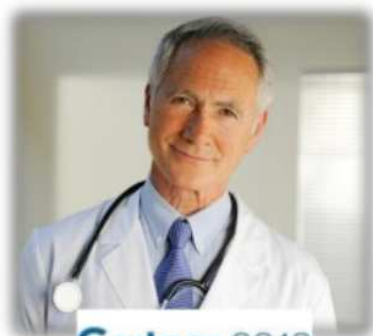
```
(:tasks-goal
  :tasks((Diagnosis Juan)
         (Treatment Juan)
         (Follow-up Juan)
        )
```

```
(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-winning technology**



**Gartner 2012  
CoolVendor**

“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.



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

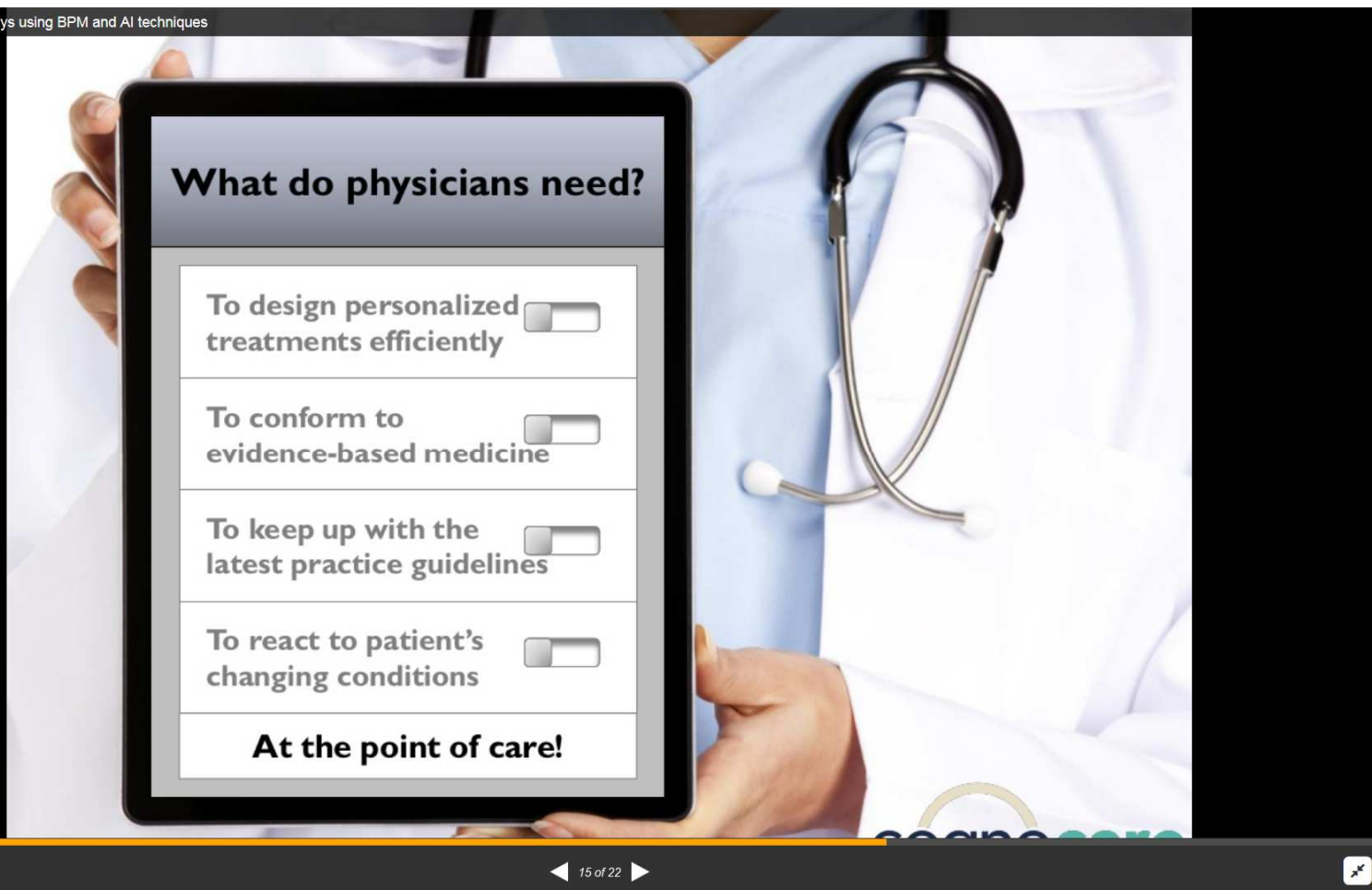


**International Conference on Planning & Scheduling. Award for Excellence in Knowledge Engineering.**  
ICAPS 2009. Tesalónica, Greece.



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

Personalized Care Pathways using BPM and AI techniques

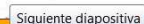


Physicians may modify the details of the treatment

### Alerts about scheduled lab tests

Detailed explanations about dosages

Next scheduled test



### No repetibilidad:

- El lenguaje textual HPDL y su versión gráfica EKDL pueden usarse como modelo declarativo de procesos.

### Impredictibilidad:

- La generación de planes dependientes de la situación, y guiada por objetivos, gestiona la impredictibilidad de los eventos

### Emergente:

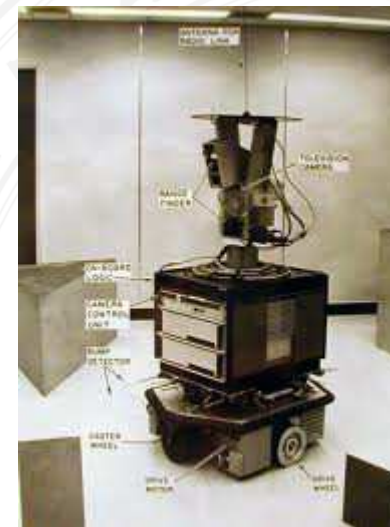
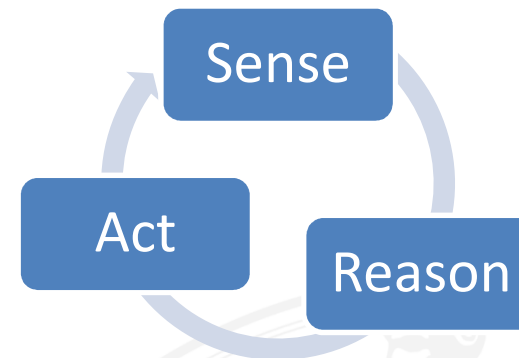
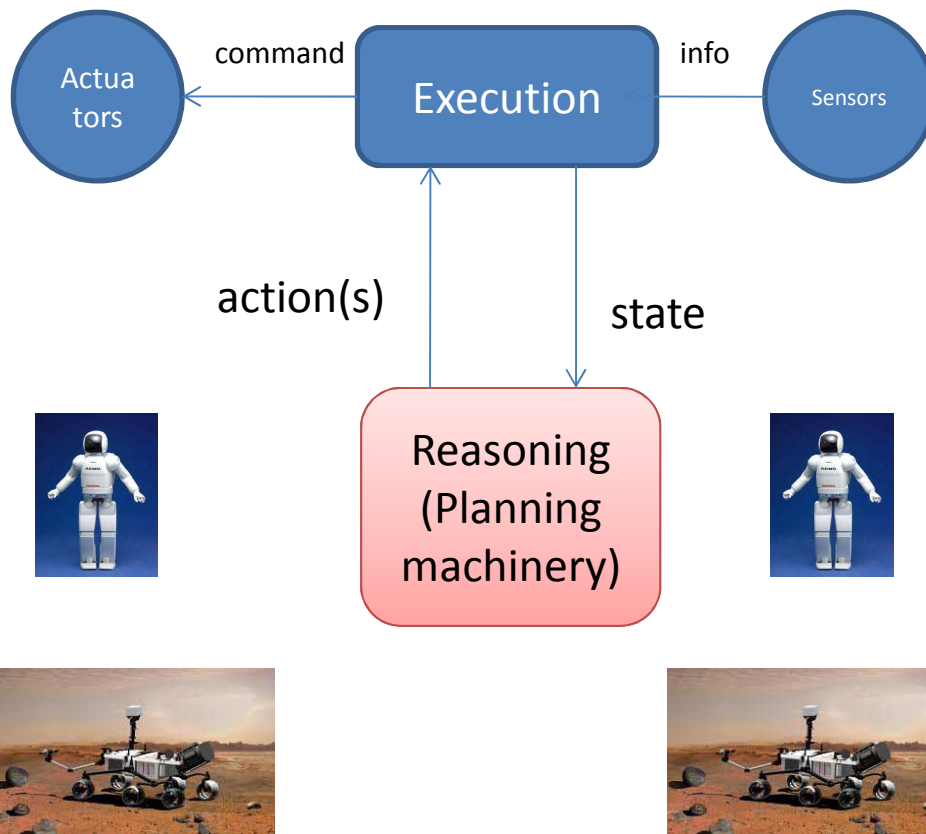
- El ciclo de planificación continua (planificar, ejecutar, detectar situación, planificar) ayuda a gestionar el despliegue incremental de un plan (planificar hasta donde se pueda, adquirir información, continuar con la planificación)

### Robustez ante situaciones variables:

- La replanificación basada en HTN permite gestionar situaciones predecibles
  - Gestión de excepciones ocurridas por eventos controlables.
- La replanificación basada en reglas de reparación permite gestionar situaciones impredecibles
  - Gestión de excepciones ocurridas por eventos exógenos (no controlables).



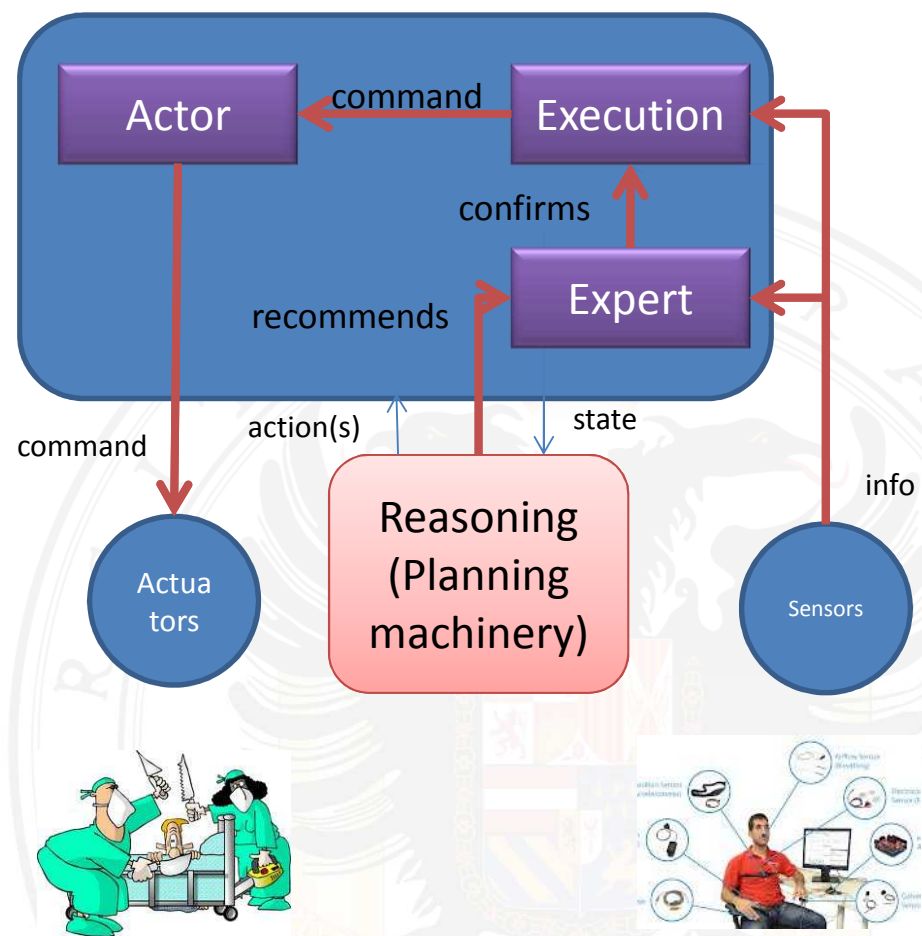
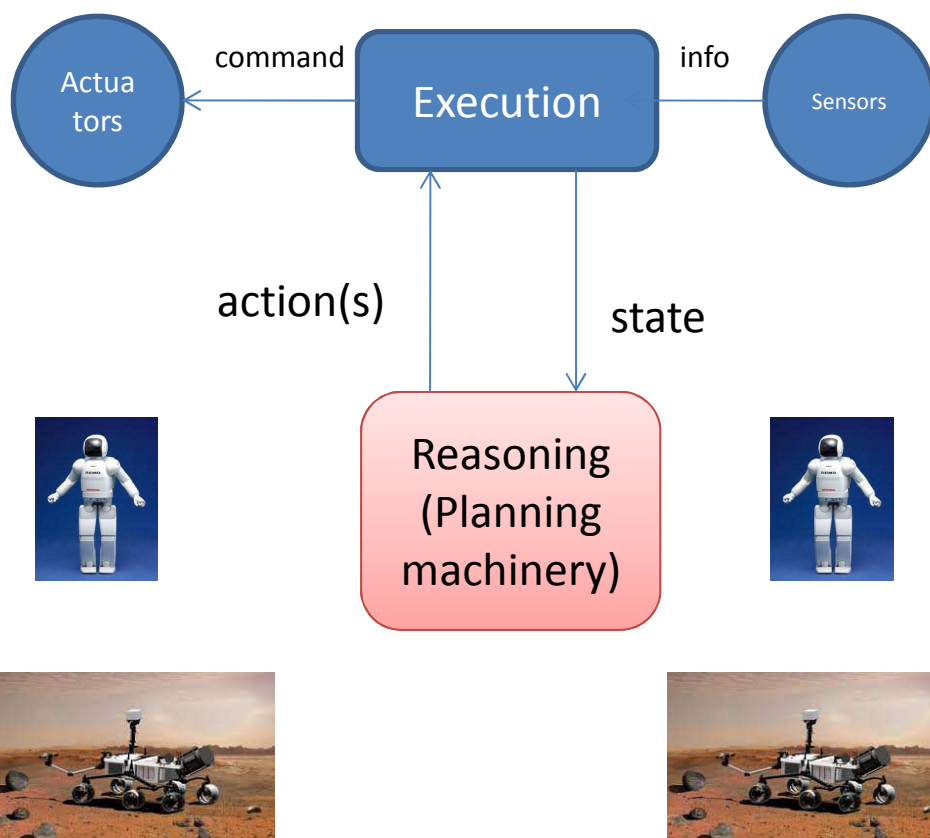
## Autonomous systems



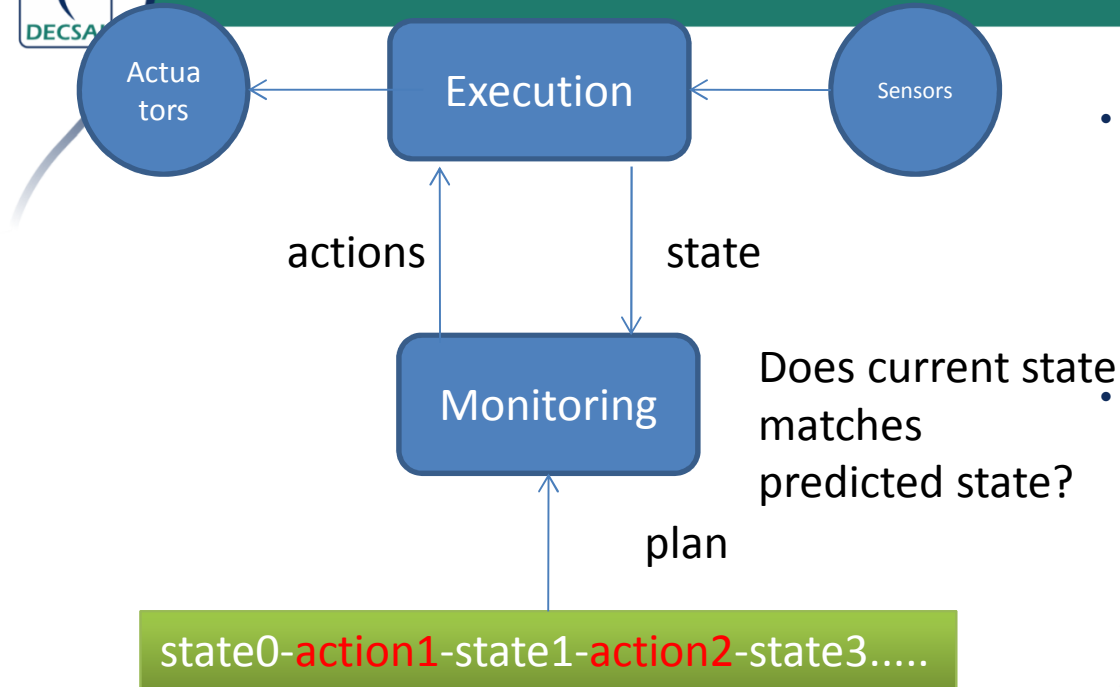
R.E. Fikes, P.E. Hart, N.J. Nilsson. **Learning and executing generalized robot plans.** *Artificial Intelligence*, 3, 1972, 251-288.

## Autonomous systems

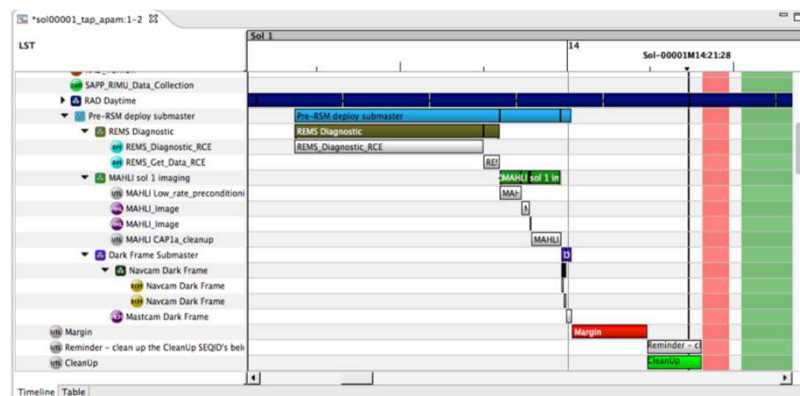
## Intelligent tools/Intelligent Process Management



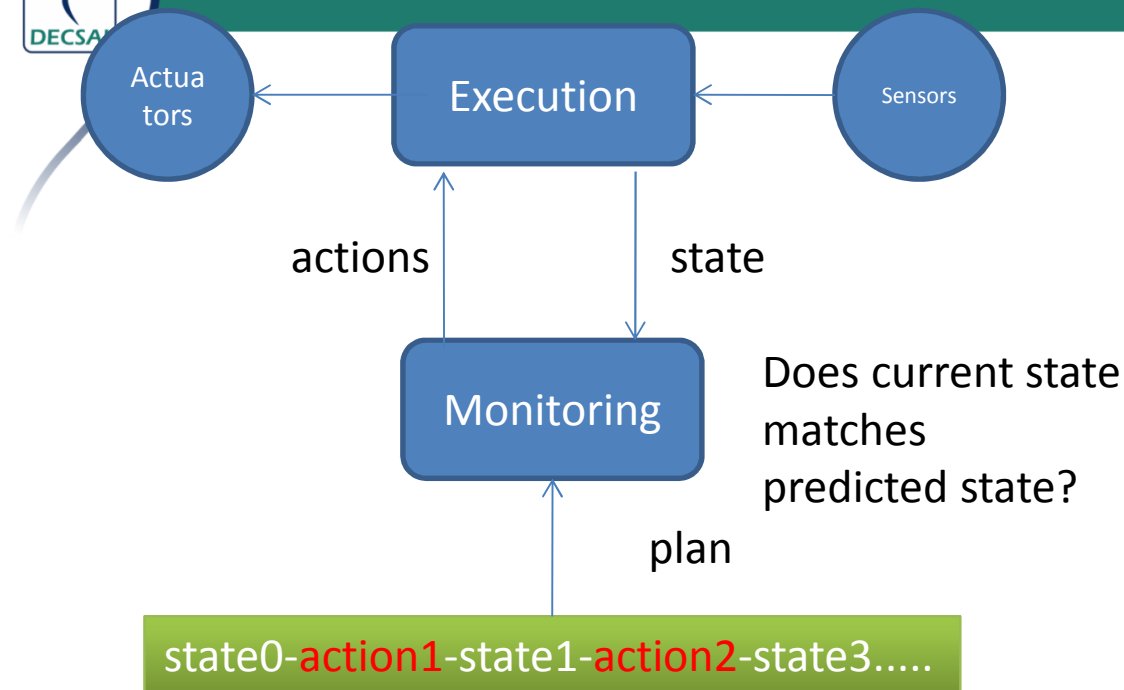




- A key issue when executing actions is to test if either a given action can be executed on the current state of the world or its effects have been successfully achieved in the world.
- This is carried out by a monitoring process that receives as input a plan, receives information about the current world state, decides which actions have to be executed, keeps track of executed actions, and



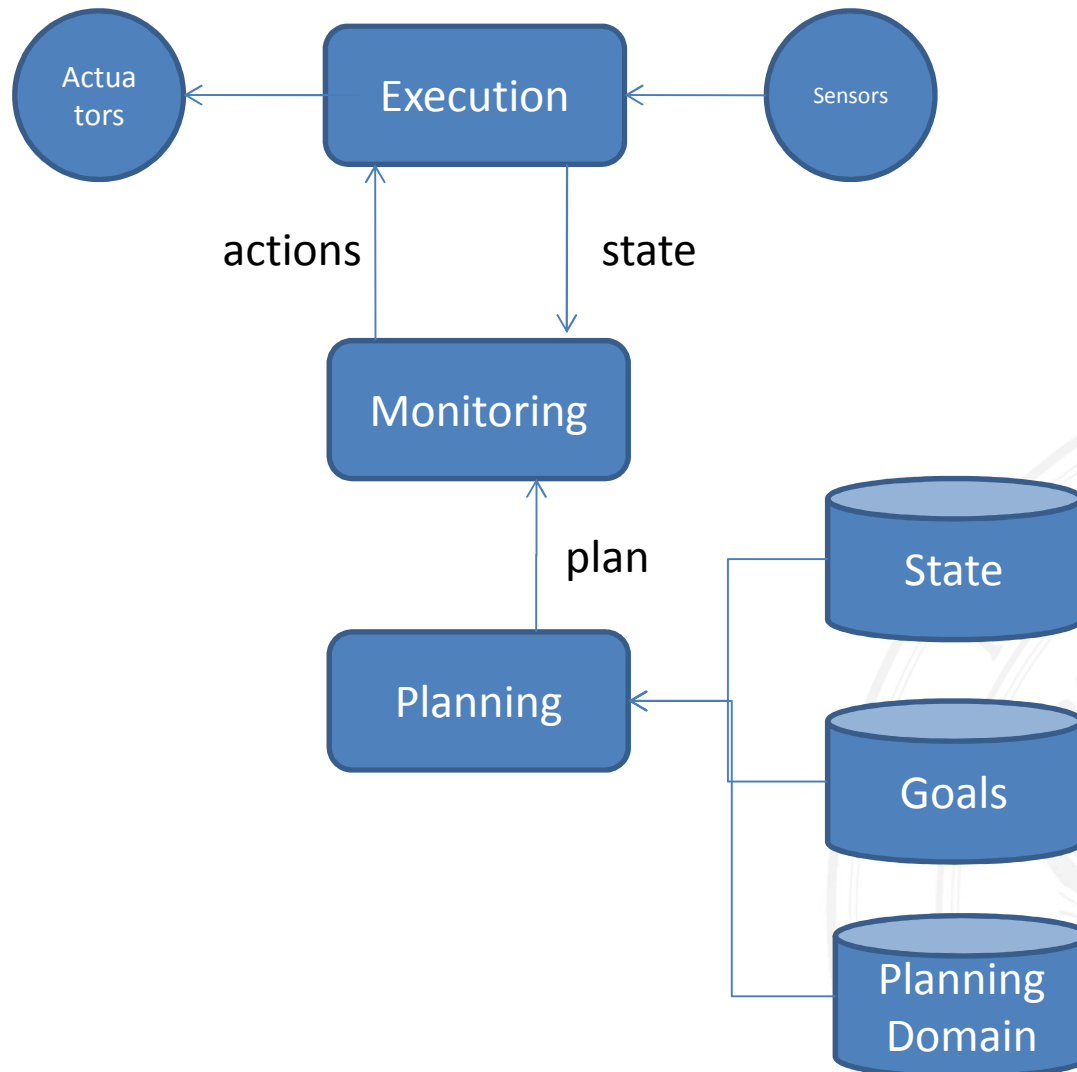
Start	End	Dur (hrs.)	Task	Resource
07/12/2007	08/12/2007	24	Previous Eval.	Tomas
08/12/2007	23/12/2007	360	Summary OPPIA	
08/12/2007	09/12/2007	24	AdminDrug	VCR
15/12/2007	16/12/2007	24	AdminDrug	VCR
22/12/2007	23/12/2007	24	AdminDrug	VCR
08/12/2007	23/12/2007	360	AdminDrug	PRD
08/12/2007	23/12/2007	360	AdminDrug	PRC
08/12/2007	09/12/2007	24	AdminDrug	ADR
22/12/2007	23/12/2007	24	AdminDrug	ADR
07/01/2008	08/01/2008	24	Previous Eval.	Juan
08/01/2008	23/01/2008	360	Summary OPPIA	
08/01/2008	09/01/2008	24	AdminDrug	VCR
15/01/2008	16/01/2008	24	AdminDrug	VCR
22/01/2008	23/01/2008	24	AdminDrug	VCR
08/01/2008	23/01/2008	360	AdminDrug	PRD
08/01/2008	23/01/2008	360	AdminDrug	PRC
08/01/2008	09/01/2008	24	AdminDrug	ADR
22/01/2008	23/01/2008	24	AdminDrug	ADR



A key issue when executing actions is to test if either a given action can be executed on the current state of the world or its effects have been successfully achieved in the world.

This is carried out by a monitoring process that receives as input a plan, decides which actions have to be executed, keeps track of executed actions, and receives information about the current world state.

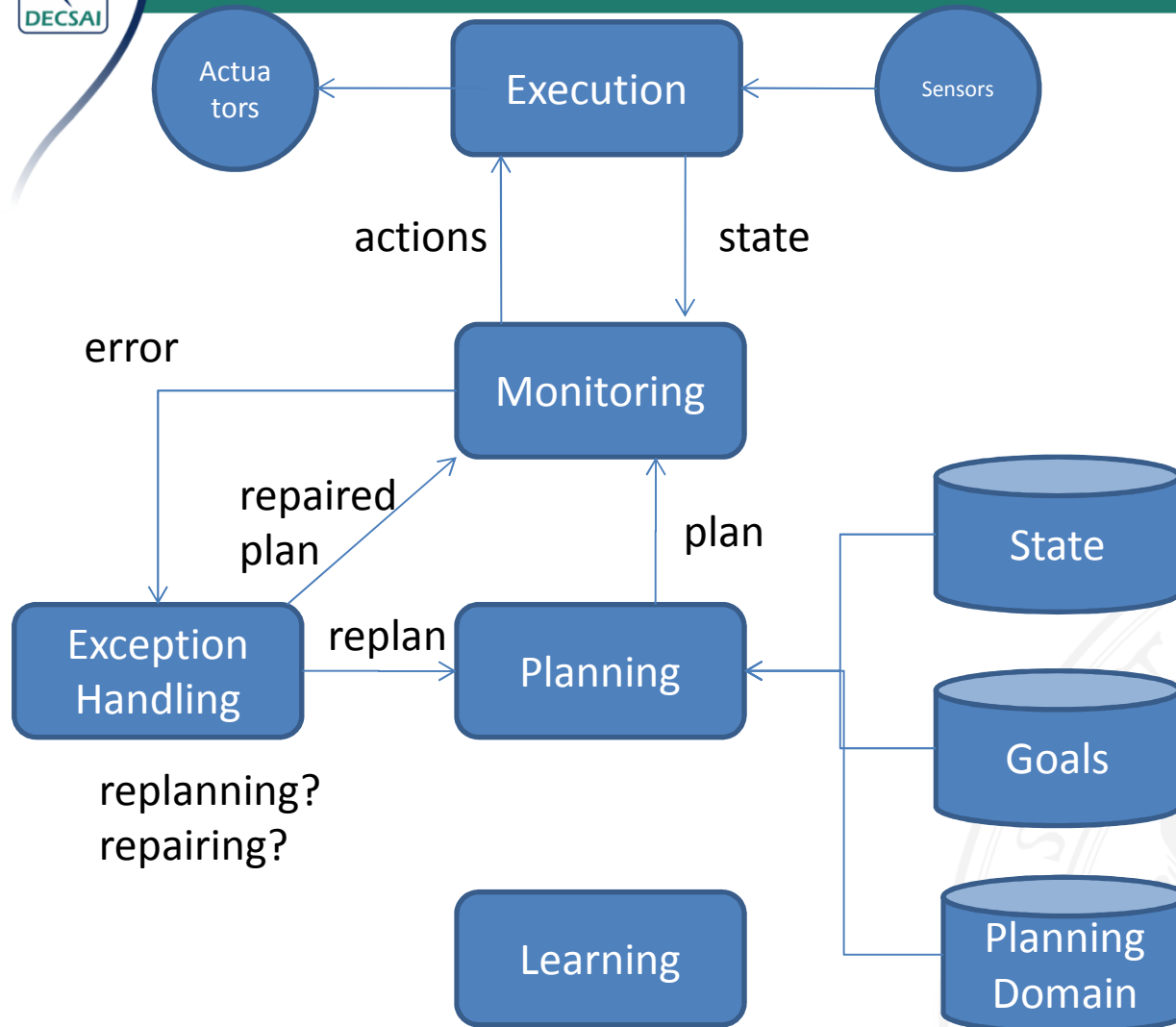
1. C. Fritz, S.A. McIlraith. **Monitoring plan optimality during execution**. *Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS)*, 2007, 144-151.
2. M.M. Veloso, M.E. Pollack, M.T. Cox. **Rationale-based Monitoring for Planning in Dynamic Environments**. *Proceedings of the 4<sup>th</sup> International Conference on Artificial Intelligence Planning Systems (AIPS)*, 1998.
3. M. Ai-Chang, J. Bresina, L. Charest, A. Chase, J.-J. Hsu, A. Jonsson, B. Kanefsky, P. Morris, K. Rajan, y J. Yglesias, «Mapgen: mixed-initiative planning and scheduling for the mars exploration rover mission», *Intelligent Systems, IEEE*, vol. 19, n.º 1, pp. 8–12, 2004.
4. G. Milla-Millán, J. Fdez-Olivares, I. Sánchez-Garzón, D. Prior, y L. Castillo, «Knowledge-Driven Adaptive Execution of Care Pathways Based on Continuous Planning Techniques», in *ProHealth 2012/KR4HC 2012*, Springer, 2012.
5. 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.



In order to reach a fully automated behaviour, action plans need to be automatically generated upon three basic inputs: current world state, a planning domain and a set of goals.

The current state provides information about what there is in the world in a given instant, the planning domain information about what objects there are in the world and how actions affect the objects, and the goals say which state should be reached with the plan.

All this stuff is knowledge, so there is also a need for knowledge representation.



In order to obtain a robust autonomous behaviour, since the world is dynamic and uncertain, planning applications have to cope with unexpected events that invalidate the current plan.

From an engineering p.o.v, the monitoring has to raise errors that are managed by an exception handling process. Basically, it decides whether to replan (build a new plan from the current state) or repair the plan.

M. Fox, A. Gerevini, D. Long, y I. Serina, «Plan stability: Replanning versus plan repair», in *Proceeding of ICAPS06*, 2006, pp. 212-221.

K. L. Myers, «CPEF: A Continuous Planning and Execution Framework», *AI Magazine*, vol. 20, n.º 4, p. 63, dic. 1999.

- Developing AI Planning based applications requires to develop several components.
- It is still an art, but several architectures have been proposed.

[1] PRS

M. P. Georgeff y A. L. Lansky, «Reactive reasoning and planning», in *Proceedings of the sixth national conference on artificial intelligence (AAAI-87)*, 1987, vol. 677682.

[2] TCA

R. G. Simmons, «Concurrent planning and execution for autonomous robots», *Control Systems, IEEE*, vol. 12, n.º 1, pp. 46–50, 1992.

[4] IxTET

M. Ghallab y H. Laruelle, «Representation and control in IxTeT, a temporal planner», in *Proceedings of AIPS*, 1994, vol. 94, pp. 61–67.

[5] IDEA

P. Aschwanden, V. Baskaran, S. Bernardini, C. Fry, M. Moreno, N. Muscettola, C. Plaunt, D. Rijsman, y P. Tompkins, «Model-unified planning and execution for distributed autonomous system control», in *AAAI Fall Symposium on Spacecraft Autonomy*, 2006.

[6] APSI

A. Cesta, G. Cortellessa, S. Fratini, y A. Oddi, «Developing an end-to-end planning application from a timeline representation framework», in *21st Applications of Artificial Intelligence Conference*, 2009.

[7] PELEA

•E. Quintero, V. Alcázar, D. Borrajo, J. Fernández-Olivares, F. Fernández, A. G. Olaya, C. Guzman, E. Onaindia, y D. Prior, «Autonomous Mobile Robot Control and Learning with the PELEA Architecture», in *Automated Action Planning for Autonomous Mobile Robots*, 2011.

•Cesar Guzman, Vidal Alcázar, David Prior, Eva Onaindia, Daniel Borrajo, Juan Fdez-Olivares : **Building a Domain-Independent Architecture for Planning, Learning and Execution (PELEA)** ICAPS 2011 [System Demonstrations and Exhibits](#). Freiburg, Germany. June 2011.



How can I improve it?

Is everything effective?

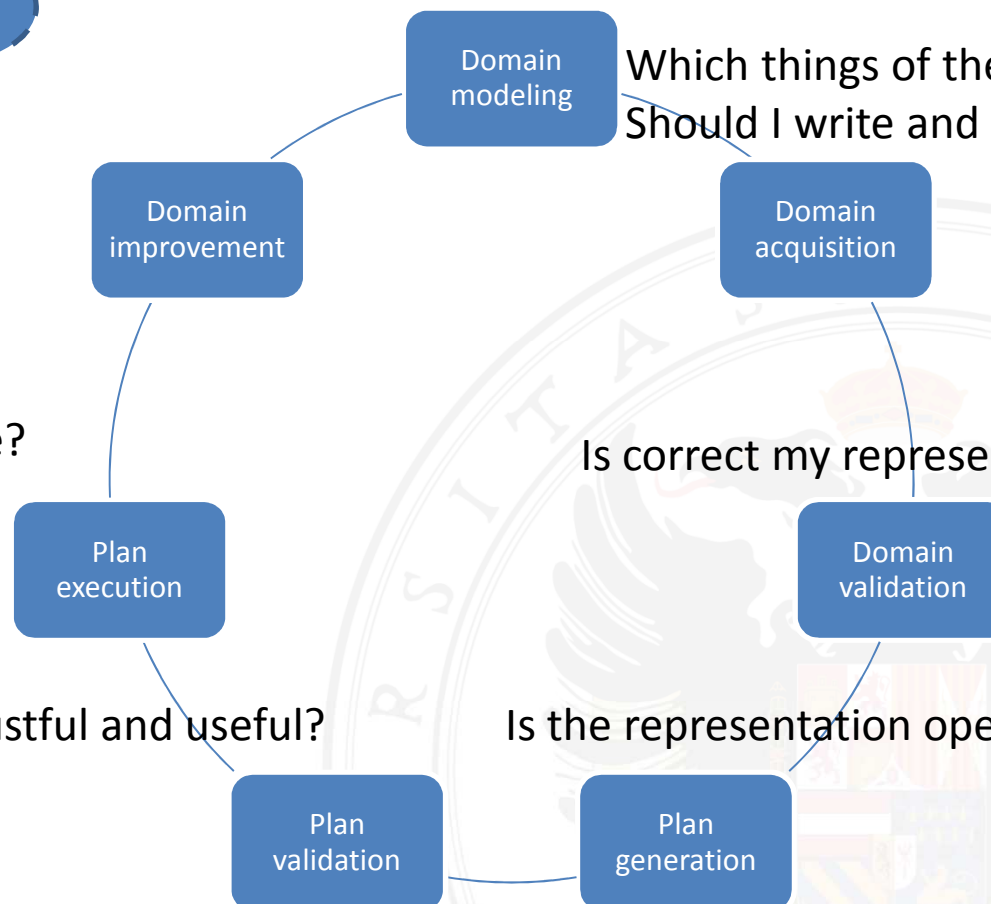
Are the results trustful and useful?

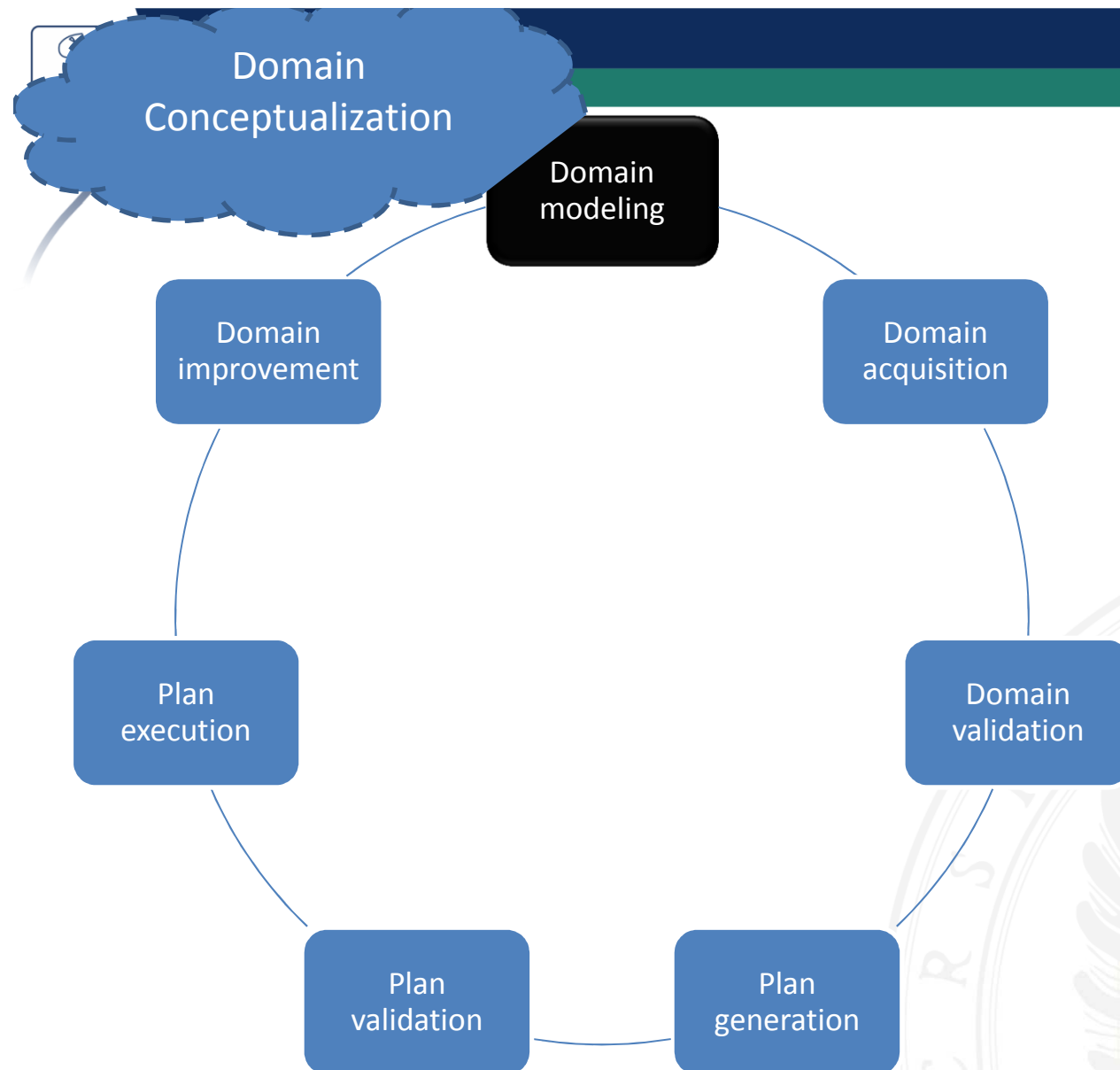
Which language should I use?

Which things of the world  
Should I write and how to do it?

Is correct my representation?

Is the representation operational?

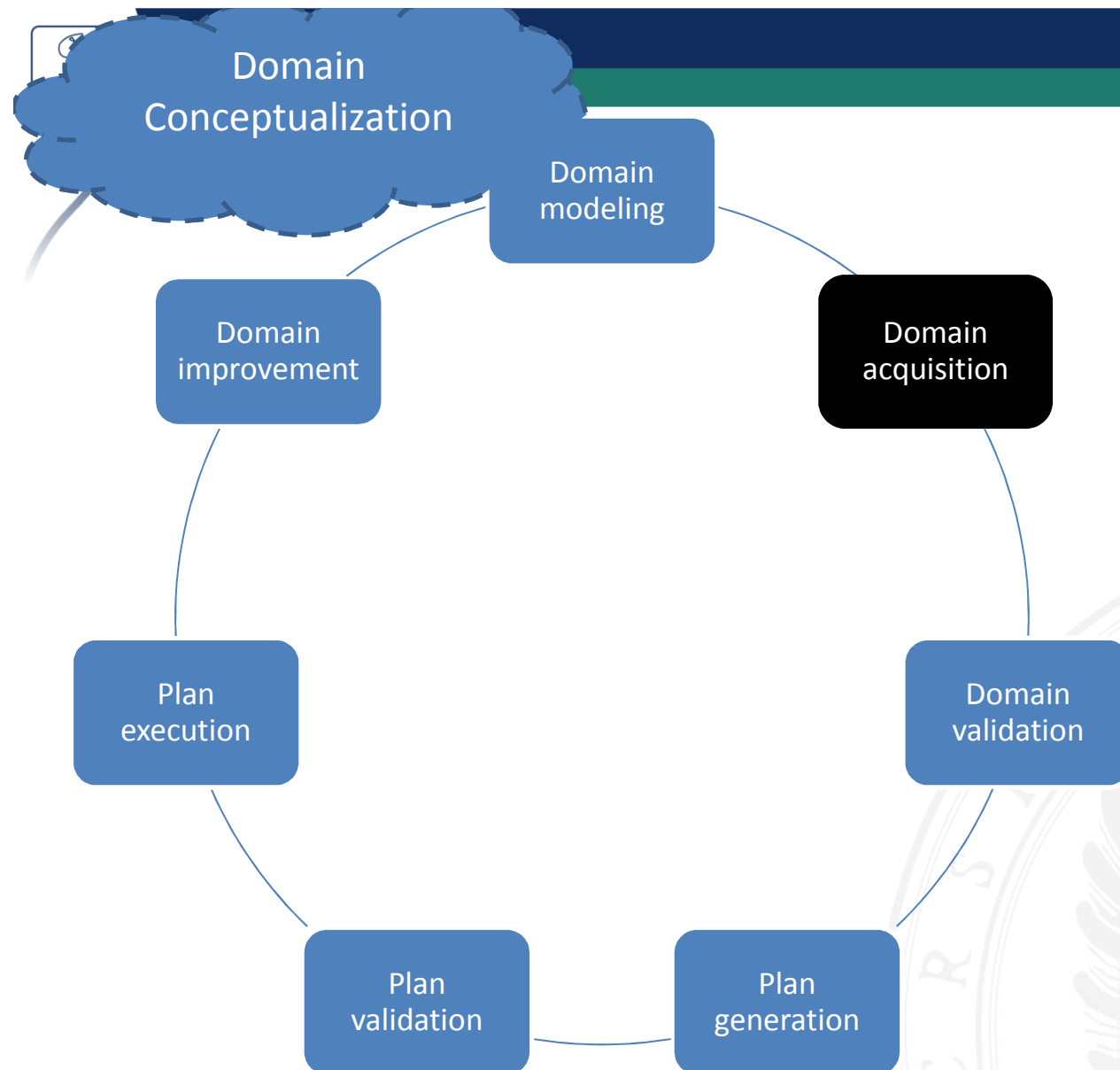




## AI Planning Life-Cycle

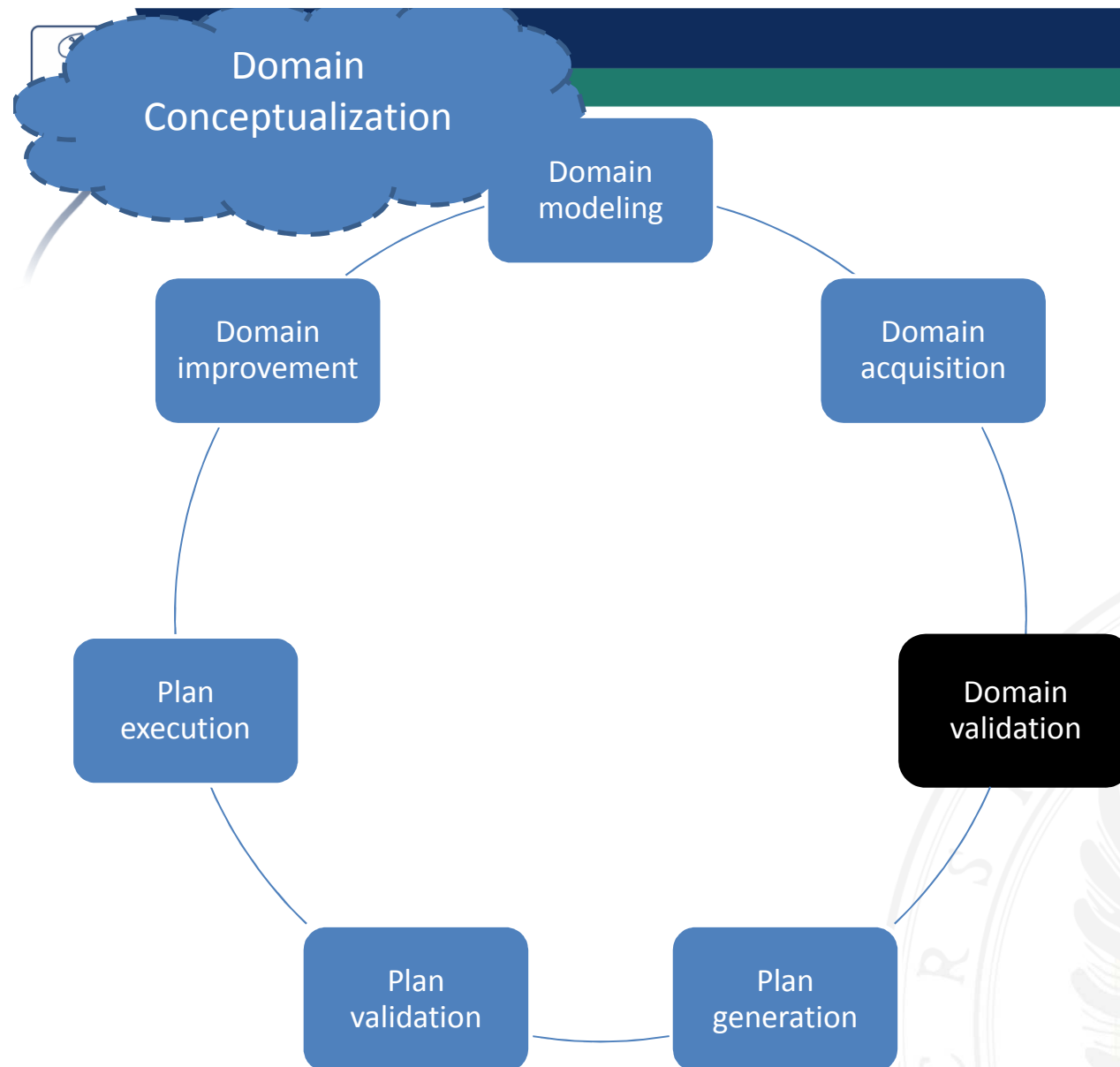
The first step consists on **transforming the real world into concepts**. The nature and structure of such concepts are necessarily affected by the **planning language** that we are going to use. There are **many planning languages**, each one is defined according to a vision of the world. We may either analyze the characteristics of the domain and then decide which language use or (the most common way) first decide a language to use and then conceptualize the domain.





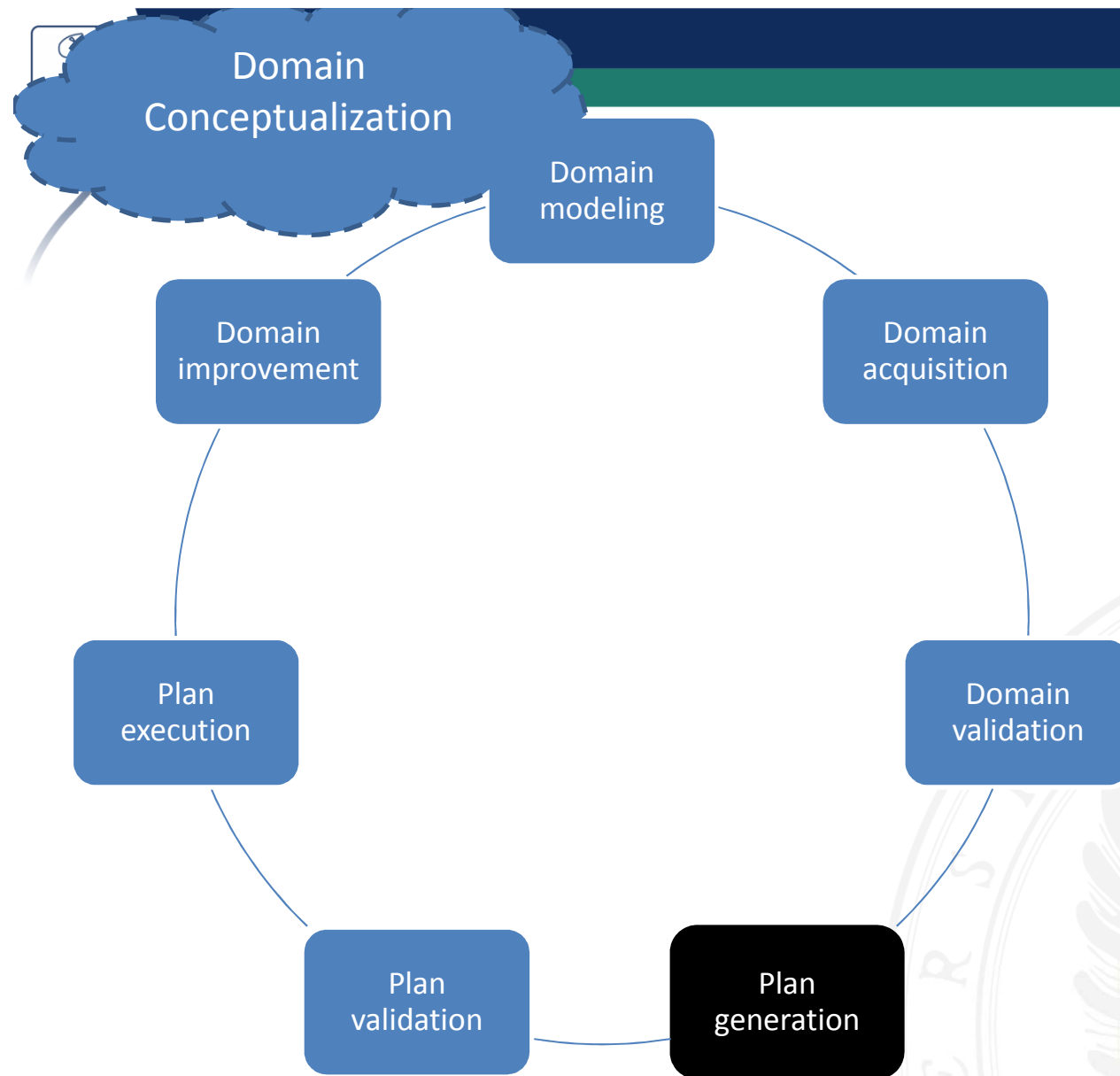
## AI Planning Life-Cycle

The next step is a process that result in a “domain description”, that is, **planning knowledge**, a planning domain that comprises objects, properties, relationships, actions, tasks, constraints, etc. And problems (instances of objects and goals). These symbolic representations are already **interpretable by a planner**, and, therefore, we are able to obtain plans.



## AI Planning Life-Cycle

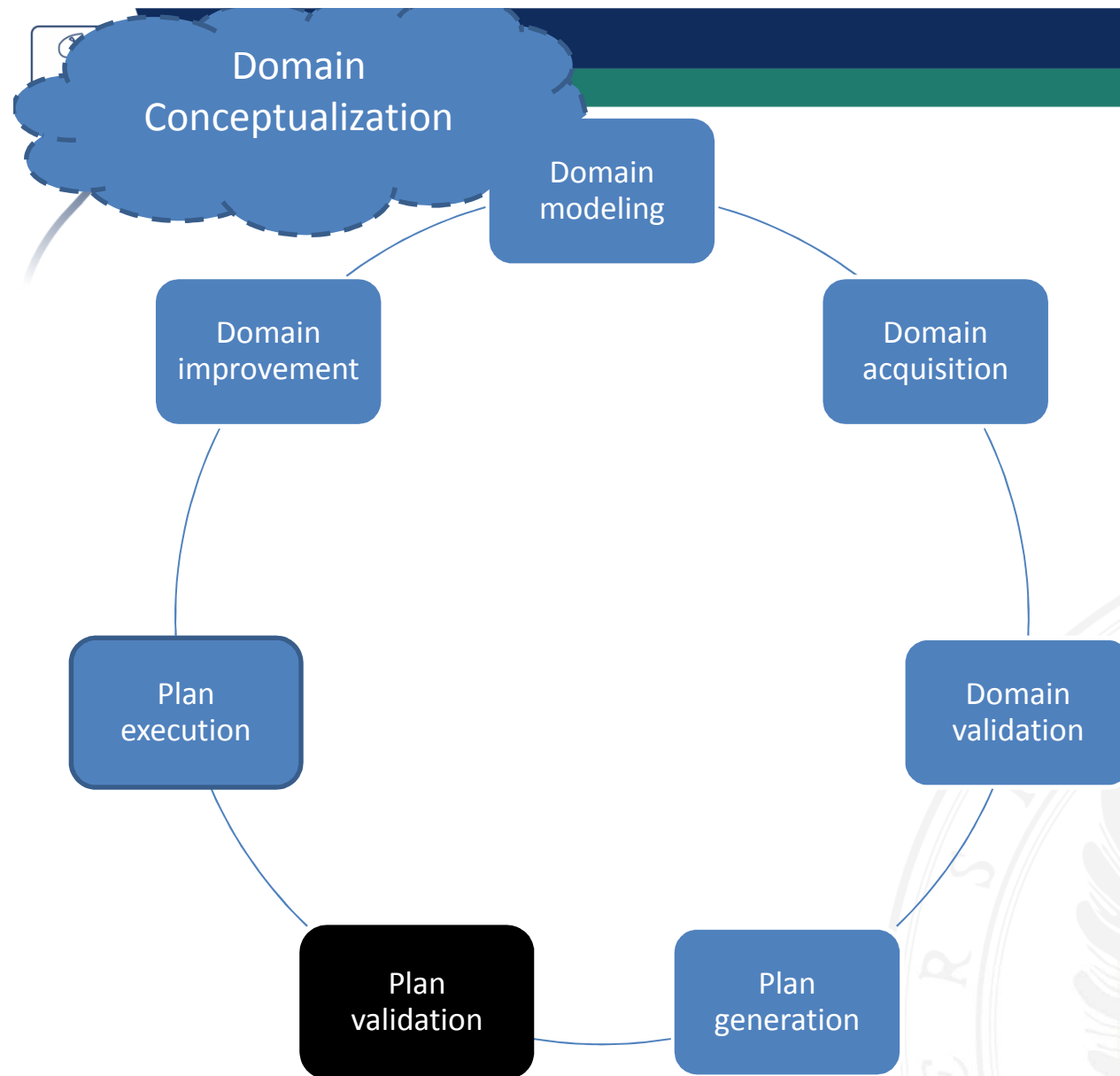
But, are we sure that the **domain is correct** and also the plans obtained?. Like in programming, we have not evidence that our planning domain is correct unless we test it. We can use a **handcraft** approach or a more robust one (based on **formal criteria**). If we validate a domain by hand, we have very few arguments to convince an eventual user.



## AI Planning Life-Cycle

The language used limits the planning techniques to be used in the planner. We cannot use **any language with any planner**.

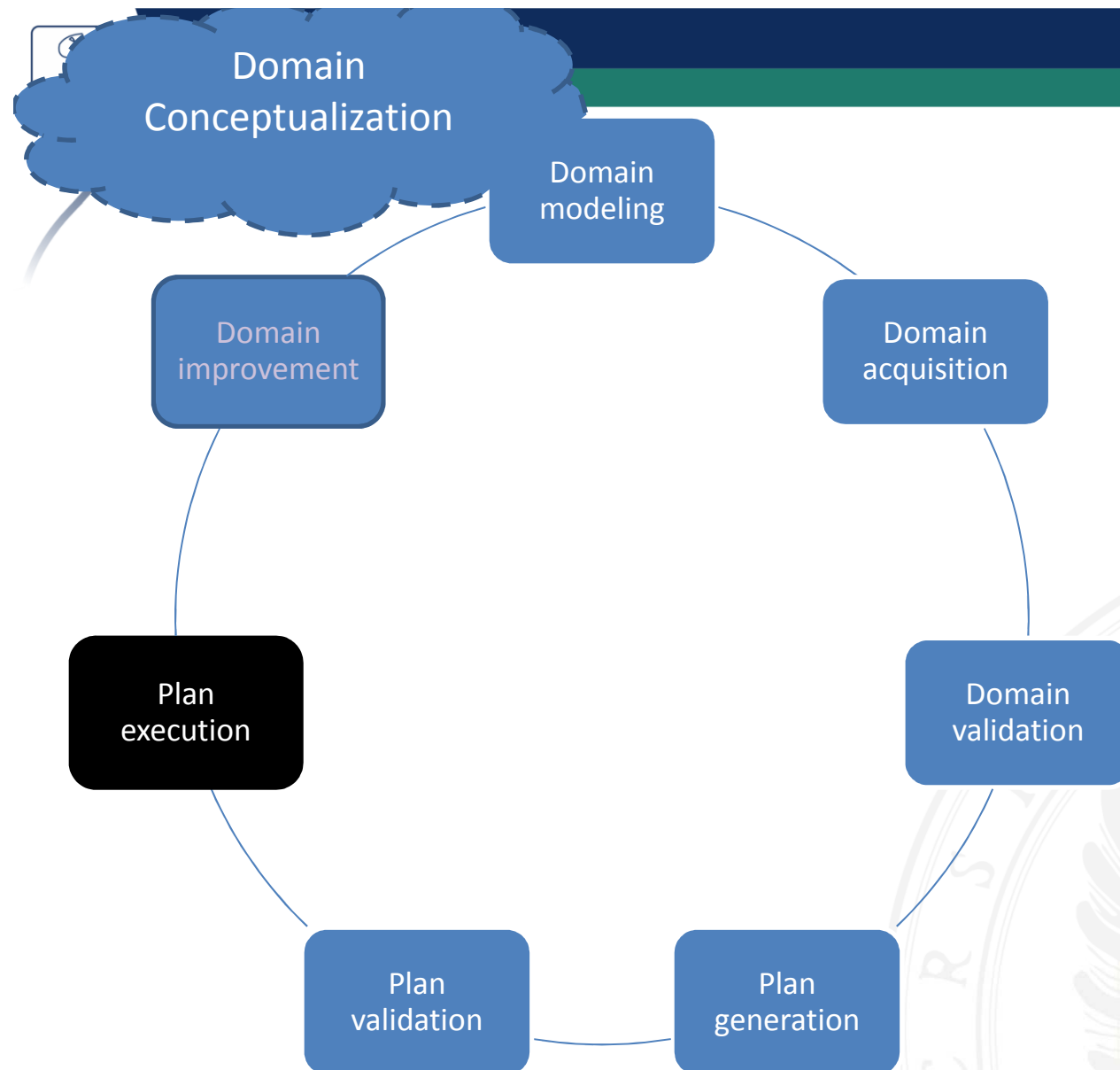
Both decisions (the language to use and the planner to operate with) are interdependent.



## AI Planning Life-Cycle

Plans are obtained for a huge amount of possible contexts. Why should the **user trust every plan?**

Again we are forced to provide support to validate plans. The more usable is this step, the more confident the user will be with the results obtained.



## AI Planning Life-Cycle

What can we do with a validated plan, obtained by a robust planner from a valid domain?

Plans are sequences of actions that describe what, under which conditions, and when to do something in order to achieve a goal or perform a task.

The next natural step is to execute the plan.

