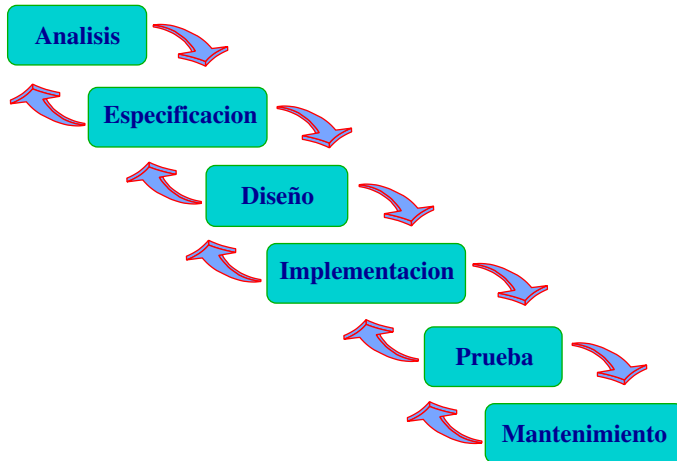


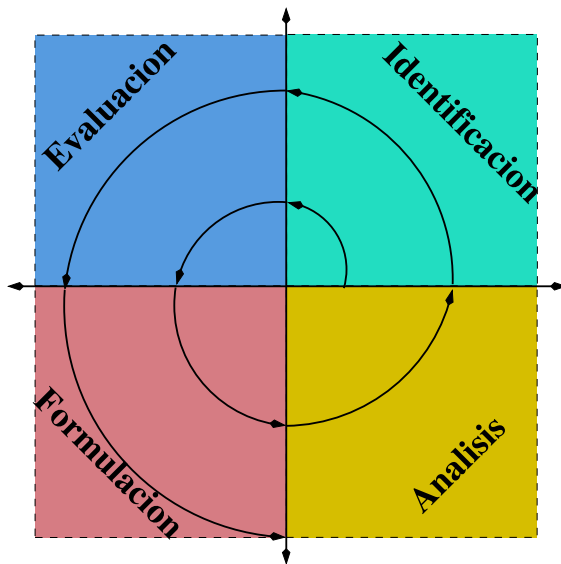
# Desarrollo de SBC

- El punto más importante del desarrollo de SBC es la extracción del conocimiento
- Requiere la interacción entre el **Ingeniero del Conocimiento** y el experto
- Las metodologías de ingeniería de software han de encajar este proceso entre sus fases
- Las metodologías de ingeniería del software han de adaptarse a las características específicas de los SBC

# IS: Modelo en cascada



# IS: Modelo en espiral



# Diferencias de los SBC

- Sistemas software convencionales  $\Rightarrow$  Algoritmos conocidos y de uso común
- SBC  $\Rightarrow$  Conocimiento incompleto, impreciso, heurístico
- Sistemas software convencionales  $\Rightarrow$  Posible estimar la naturaleza y cantidad del conocimiento
- SBC  $\Rightarrow$  Difícil estimar la naturaleza y cantidad del conocimiento

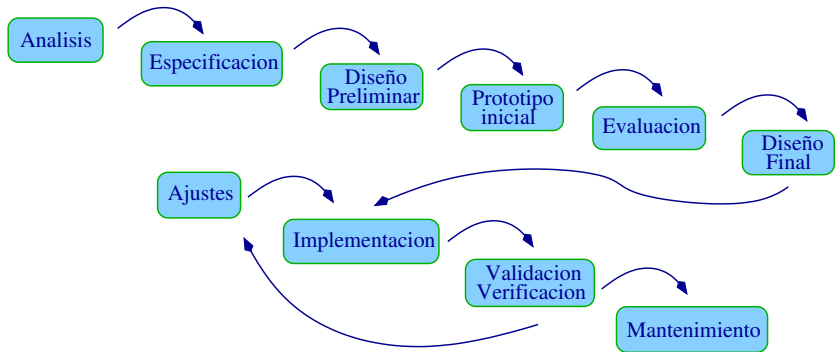
# Diferencias de los SBC

- Es complicado obtener un diseño adecuado en las fases iniciales
- Decisiones iniciales erróneas pueden provocar el replanteamiento radical del diseño durante el desarrollo
- El **ingeniero del conocimiento** debe realizar un proceso de adquisición del conocimiento  $\implies$  Entrevistas con los expertos
  - El IC debe aprender los elementos básicos del dominio
  - Encontrar un formalismo representación que pueda entender el experto
  - Los expertos prefieren casos al razonamiento a partir de definiciones generales
  - A los expertos les es difícil explicitar su conocimiento en detalle (paradoja del experto)

# Diferencias de los SBC

- Solución: **Diseño incremental y prototipado rápido**
- Objetivo: Desarrollar un prototipo funcional que recoja las funcionalidades básicas del sistema
- El análisis y la especificación deben tener en cuenta el sistema completo
- El diseño e implementación se limita al prototipo inicial
- Este prototipo se completa incrementalmente
- Ventaja: Disponemos de un sistema funcional durante todo el proceso

# Ciclo de vida de un SBC



# Ciclo de vida de un SBC (I)

1. **Análisis del problema:** Recopilar información sobre el proyecto y determinar su viabilidad.
2. **Especificación de requerimientos:** Fijar los objetivos y métodos para conseguirlos.
3. **Diseño preliminar:** Decisiones a alto nivel sobre el diseño (formalismo de representación del conocimiento, herramientas, fuentes de conocimiento)
4. **Prototipo Inicial y evaluación:** Construir un prototipo con cobertura limitada, evaluar las decisiones de diseño a partir del prototipo
5. **Diseño final:** Validar las decisiones y proponer el diseño del sistema de manera que permita un desarrollo incremental.



## Ciclo de vida de un SBC (II)

6. **Implementación:** Completar la adquisición del conocimiento, ampliar incrementalmente el prototipo inicial.
7. **Validación y verificación:** Comprobar que el sistema cumple las especificaciones.
8. **Ajustes de diseño:** Realimentar el proceso (los cambios en el diseño deberían ser mínimos)
9. **Mantenimiento:** Mantener el sistema.

# Metodologías especializadas

## ● CommonKADS

- Ciclo de vida en espiral y modelado mediante herramientas parecidas a UML
- Se construyen seis modelos: Organización, tareas, agentes, comunicación, conocimiento y diseño.

## ● MIKE

- Ciclo de vida en espiral: Adquisición del conocimiento (modelo de adquisición y modelo de estructura), diseño, implementación, evaluación.

# Una metodología simplificada

- Para aplicaciones pequeñas se puede aplicar una metodología en cascada que integra todo el proceso de desarrollo
  - 1 **Identificación del problema**
  - 2 **Conceptualización**
  - 3 **Formalización**
  - 4 **Implementación**
  - 5 **Validación y Prueba**

# Identificación

- Debemos determinar si el problema es adecuado
  - ¿Hay una solución algorítmica?
  - ¿Disponemos de fuentes de conocimiento?
  - ¿El tamaño/objetivo/complexidad del problema es adecuado?
- Buscar y evaluar las fuentes de conocimiento
- Determinar el conocimiento necesario para el sistema
- Establecer los objetivos del sistema (¿Que respuesta esperamos?)

# Conceptualización

Esta fase nos debería dar la perspectiva del problema desde el punto de vista del experto

- Deberemos:
  - Detallar los elementos del dominio  $\implies$  Descripción informal de la ontología
  - Descomponer el problema en subproblemas mediante refinamientos sucesivos, descubriendo los bloques de razonamiento
  - Detallar el flujo de razonamiento y las entradas y salidas de cada subproblema
  - Detallar y distinguir entre evidencias, hipótesis y acciones y descubrir sus relaciones
- Toda esta información la obtendremos a partir de la interacción con el experto (entrevistas) y las fuentes de conocimiento
- El resultado será un modelo semiformal del dominio y de los problemas y métodos de resolución

# Formalización

Esta fase transformará la perspectiva del experto en la perspectiva del ingeniero del conocimiento

- Decidir el formalismo de representación del conocimiento adecuado
- Identificar el espacio de búsqueda
- Analizar la tipología de los problemas y bloques de razonamiento y decidir los métodos de resolución de problemas adecuados
- Analizar la necesidad de tratamiento de incertidumbre y/o información incompleta

# Implementación

- Construir una ontología del dominio
- Encajar los problemas identificados en las metodologías de resolución de problemas escogidas
- Construir los diferentes módulos que correspondan a cada problema siguiendo el conocimiento obtenido
- Si utilizamos una aproximación basada en prototipado rápido construiremos el prototipo inicial y lo iremos aumentando incrementalmente

# Validación y Prueba

- Escoger casos representativos y resolverlos mediante el sistema
- Los casos deberían incluir tanto casos usados para la construcción del sistema como casos nuevos
- Si seguimos una estrategia de construcción incremental esta fase se irá repitiendo a medida que se desarrolle el prototipo
- La validación de SBC es más compleja que la de los sistemas de software habituales



# Tipos de problemas en SBC

- La identificación de tipologías de problemas en SBC facilita su desarrollo
- Cada tipo permite determinar:
  - Un conjunto de tareas usuales
  - Un conjunto de metodologías de resolución específicas
  - Métodos de representación del conocimiento e inferencia adecuados
- Clasificaremos los problemas a resolver en un SBC a partir de dos tareas genéricas:
  - **Tareas de análisis:** Interpretación de un sistema
  - **Tareas de síntesis:** Construcción de un sistema

# Síntesis - Análisis

Ambas tareas se pueden especializar

- Tareas de análisis

- Identificación, nos dice que tipo de sistema tenemos
  - Monitorización, detecta discrepancias de comportamiento
  - Diagnóstico, explica las discrepancias
- Predicción, nos dice que salida podemos esperar
- Control, determina que entradas nos permiten obtener cierta salida

- Tareas de síntesis

- Especificación, busca las restricciones que se deben satisfacer
- Diseño, genera una configuración de elementos según unas restricciones
- Ensamblaje, construye un sistema juntando elementos

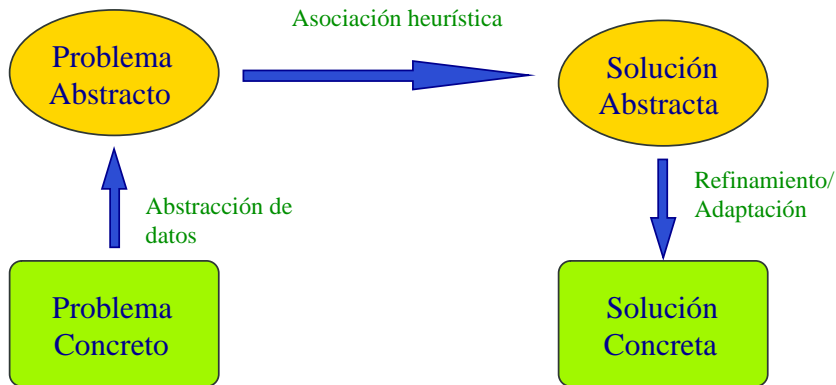
# Métodos de resolución de problemas

- Cada tipo de problema genérico tiene unas características especiales
- Existen métodos de resolución adecuados para cada tipo
- Nos centraremos en dos métodos:
  - Clasificación Heurística (*Heuristic Classification*)
  - Resolución Constructiva (*Constructive Problem Solving*)

# Clasificación Heurística

- Es adecuado para tareas de **análisis**
- El objetivo es escoger una solución de entre un conjunto limitado
- Asociamos los datos de entrada con las soluciones (simple o mediante razonamiento)
- Tres fases:
  - 1 **Abstracción de datos** (Definicional, cualitativa, generalización, ...)
  - 2 **Asociación heurística**
  - 3 **Refinamiento**

# Clasificación Heurística



# Clasificación Heurística: Adquisición del conocimiento

- La adquisición del conocimiento que permite resolver un problema mediante clasificación heurística se puede hacer de forma sistemática.
- Podremos distinguir tres conjuntos de conceptos:
  - **Las hipótesis:** Soluciones posibles a nuestro problema
  - **Los síntomas:** Características que describen las hipótesis
  - **Las causas originales:** Información del problema que lleva a los síntomas
- Entre cada conjunto de conceptos deberemos extraer el conjunto de deducciones que llevan del uno al otro
- De las causas originales a los síntomas tendremos las reglas de abstracción
- De los síntomas a las hipótesis tendremos las reglas de asociación heurística

# Clasificación Heurística: Adquisición del conocimiento

- Para cada grupo de reglas deberemos:
  - Obtener qué conceptos del primer conjunto (antecedentes) están asociados con los del segundo (consecuentes)
  - Escoger como antecedentes de las reglas los conceptos que sean diferenciadores de cada consecuente (separabilidad)
  - Si es necesario se deberían añadir conceptos intermedios que lleven de los antecedentes a los consecuentes y crear las cadenas de deducción necesarias
  - Observar la confianza de las asociaciones entre antecedentes y consecuentes (incertidumbre)
- Si las hipótesis son soluciones abstractas  $\implies$  Determinar las reglas de refinamiento

# Clasificación Heurística: Ejemplo (1)

- Supongamos que queremos construir un SBC para la concesión de créditos bancarios
- El conjunto de respuestas que puede dar el sistema son limitadas (se concede/no se concede)
- El objetivo es determinar, dadas la características del cliente, si concederemos el crédito y en que condiciones, o si no lo concederemos.
- Es un problema de **análisis** que podemos resolver mediante **clasificación heurística**.



## Clasificación Heurística: Ejemplo (2)

Supongamos que una solicitud contiene la siguiente información:

- Si tiene avales bancarios.
- Si tiene familiares que puedan responder por él.
- Si tiene cuentas corrientes, casas, coches, fincas, etc. y su valoración.
- Si tiene antecedentes de morosidad.
- Si ha firmado cheques sin fondos.
- Si tiene créditos anteriores concedidos.
- Tipo de empresa que quiere crear.
- Cantidad de dinero que solicita.

## Clasificación Heurística: Ejemplo (3)

Determinamos un conjunto de características que definen las soluciones

- Apoyo financiero (Muy bueno, bueno, normal, regular, malo, muy malo)
- Bienes
- Fiabilidad de devolución
- Compromiso con el cliente
- Viabilidad de la empresa

# Clasificación Heurística: Ejemplo (4)

Decidimos un conjunto de soluciones abstractas

- Denegación
- Aceptación
- Aceptación con rebaja
- Aceptación con interés preferente

# Clasificación Heurística: Ejemplo (5)

Determinamos unas reglas que nos abstraen los datos del problema

- *si*  $\text{avales} > \text{un millón euros o tío rico}$  entonces *apoyo financiero*=bueno
- *si*  $\text{avales} < 100000 \text{ euros}$  entonces *apoyo financiero*= malo
- *si*  $\text{suma bienes} < \text{un millón}$  entonces *bienes*=malo
- *si*  $\text{suma bienes} > \text{dos millones}$  entonces *bienes*=bueno
- *si* *cheques sin fondos o moroso* entonces *fiabilidad*=muy mala
- *si* *hamburguesería o heladería* entonces *viabilidad*=normal
- *si* *grandes almacenes o proveedor de internet* entonces *viabilidad*=muy buena
- *si* *concedido crédito*  $> \text{un millón o hermano del director}$  entonces *compromiso*=bueno
- ...

## Clasificación Heurística: Ejemplo (6)

Determinamos unas reglas que asocian las características con las soluciones

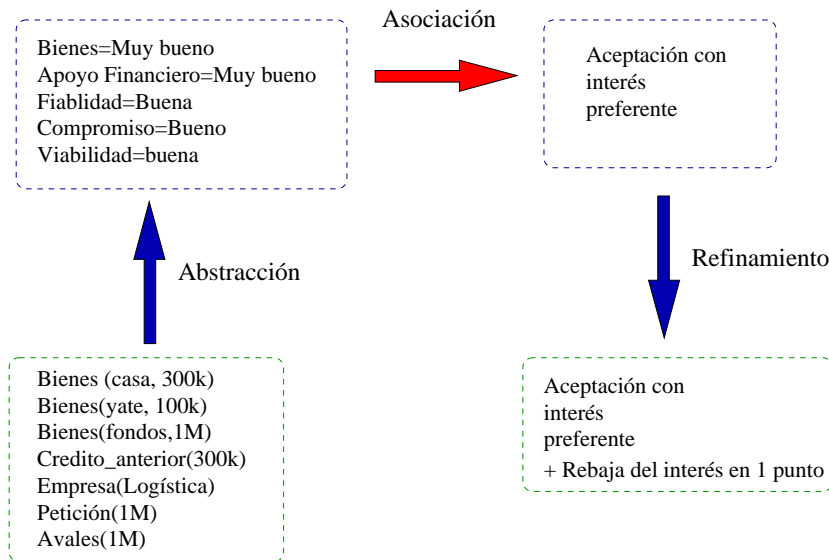
- *si apoyo financiero=regular y bienes=malo entonces denegar*
- *si fiabilidad={mala,muy mala} entonces denegar*
- *si apoyo financiero=normal y bienes=normal y viabilidad=buena entonces aceptar con rebaja*
- *si apoyo financiero=bueno y bienes=normal y compromiso=normal y viabilidad=buena entonces aceptar*
- *si apoyo financiero=bueno y bienes=bueno y compromiso=muy bueno y viabilidad=muy buena entonces aceptar con interés preferente*
- ...

# Clasificación Heurística: Ejemplo (7)

Determinamos unas reglas que refinan las soluciones que lo necesitan

- *si aceptación con rebaja y petición > 500000 euros y bienes = 500000 euros entonces rebaja a 500000 euros*
- *si aceptación con interés preferente y petición > un millón y bienes > un millón entonces rebaja de un 1 % de interés*
- *si aceptación con interés preferente y hermano del director entonces rebaja de un 2 % de interés*
- ...

# Clasificación Heurística: Ejemplo (8)



# Resolución constructiva

- Orientada a problemas en los que no existe un número acotado de soluciones
- La resolución implica construir la solución a partir de un conjunto de elementos (acciones, componentes, fallos, ...)
- Se aplica en problemas de **síntesis**
- Los métodos de búsqueda son aplicables, pero con un coste impracticable



# Resolución constructiva

- La construcción de la solución necesita conocimiento sobre:
  - El modelo de la estructura de la solución
  - El modelo del comportamiento de los elementos de la solución
  - Los operadores que permiten construir la solución
  - El conjunto de restricciones entre los elementos y la solución
  - La forma de evaluar las decisiones sobre la construcción de la solución y sobre la solución total o parcial
- Las restricciones pueden ser:
  - Sobre la configuración de los componentes (físicas, temporales, ...)
  - Respecto a las entradas/salidas/precondiciones/postcondiciones de los operadores de construcción
  - Interacciones entre los tipos de restricciones anteriores

# Métodos de resolución constructiva

- **Proponer y aplicar** (*propose and apply*): Partimos desde una solución vacía. Seleccionamos el operador de entre los posibles que nos permita extender la solución actual hasta llegar a obtener la solución completa
- **Mínimo compromiso** (*Least commitment*): Partimos desde una solución inicial completa. Seleccionamos el operador que nos permite extender la solución actual que menos restricciones nos imponga a las decisiones futuras

# Proponer y aplicar

- Buscamos en el espacio de soluciones parciales
- Partimos de una solución inicial vacía o una solución incompleta
- Cada paso va completando la solución
- Siempre elegimos el mejor operador
- Nos mantenemos en el espacio de soluciones

# Proponer y aplicar

- Necesitamos conocimiento exhaustivo sobre:
  - Operadores de resolución del problema
  - Restricciones y relaciones entre los componentes de la solución
  - Evaluación del efecto de los operadores en la solución
  - Evaluación de la bondad de la solución
- Podemos plantear la resolución de diferentes maneras
  - Construcción secuencial (necesita mucho conocimiento para ser eficiente)
  - Descomposición jerárquica de tareas (mas eficiente, pero requiere obtener operadores de descomposición)

# Proponer y aplicar: Proceso de resolución

- ➊ Inicializar el objetivo: Se crea el elemento que define la solución actual
- ➋ Proponer un operador: Se seleccionan operaciones plausibles sobre la solución actual
- ➌ Podar operadores: Se eliminan operadores de acuerdo con criterios globales.
- ➍ Evaluar operadores: Se comparan los efectos de los operadores sobre la solución y se evalúa su resultado.
- ➎ Seleccionar un operador: Se escoge el operador mejor evaluado. Si no hay operadores adecuados se reconsideran pasos anteriores
- ➏ Aplicar el operador: Se aplica el operador al estado actual.
- ➐ Evaluar el objetivo: Se para si se ha llegado al objetivo final o se reinicia el proceso.

# Mínimo Compromiso

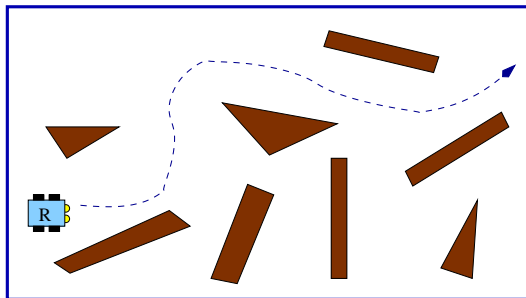
- Exploramos en el espacio de soluciones completas
- Partimos de una solución inicial completa (también puede ser no solución)
- Vamos modificando la solución mejorándola o corrigiéndola
- La elección del operador a aplicar la define la estrategia de mínimo compromiso: mínima modificación que imponga menos restricciones futuras.
- Permitimos pasar entre el espacio de soluciones y no soluciones

# Mínimo Compromiso: Proceso de resolución

- 1 Partir de una solución inicial no óptima, si es posible, que satisfaga las restricciones.
- 2 Hacer una modificación sobre la solución usando la heurística de mínimo compromiso (acción que menos restricciones imponga sobre la solución)
- 3 Si la modificación viola alguna de las restricciones deshacer alguno de los pasos anteriores haciendo las mínimas modificaciones (Puede no ser el último paso)

# Resolución Constructiva: Ejemplo (1)

- Queremos planificar la mejor trayectoria de un robot en una habitación
- La habitación tiene un conjunto de obstáculos que queremos evitar
- Disponemos de un conjunto de operadores:
  - Movernos hacia adelante o hacia atrás a cierta velocidad cierta distancia
  - Girar cierto número de grados





# Resolución Constructiva: Ejemplo (2)

- Restricciones globales: Llegar a la puerta de salida, trayectoria mínima en recorrido y tiempo
- Restricciones de elección de operadores: No chocar con obstáculos o la pared, mantener la distancia para poder maniobrar
- Evaluación de los operadores:
  - Mover: Mejor cuanto más lejos y más deprisa nos lleve al objetivo
  - Girar: Mejor cuanto mas lejos deje los obstáculos de nuestra trayectoria