

Resolución general de problemas

- Los métodos de resolución de problemas que hemos visto son de aplicación general
- Se fundamentan en una función heurística para obtener el orden de exploración de soluciones
- La capacidad expresiva de los heurísticos es reducida
- Una única función no puede representar todas las decisiones de exploración en el problema
- El ahorro en coste computacional es limitado
- Con conocimiento más específico se podrían tomar mejores decisiones

De los Sistemas Expertos a los SBC

Sistemas Expertos

- El objetivo es emular la capacidad de resolución de expertos humanos
- Se construyen por procesos de ingeniería del conocimiento
- Basados principalmente en sistemas de reglas de producción
- Sistemas cerrados con poca capacidad de aprendizaje

Sistemas Basados en el Conocimiento

- El objetivo es usar conocimiento del dominio para solucionar problemas
- Incluyen procesos automáticos de adquisición del conocimiento al proceso de ingeniería del conocimiento
- Metodologías y arquitecturas heterogéneas (reglas, casos, modelos cualitativos, agentes inteligentes, computación emergente, ...)
- Sistemas adaptables con capacidad de aprendizaje

Características de los SBC

- Los SBC se aplican a problemas complejos donde los sistemas convencionales de software no son suficientes.
- La naturaleza de estos problemas requieren ciertas características:
 - Flexibilidad para abordar diferentes problemas
 - Emulación de comportamiento racional como mecanismo de resolución
 - Operar en un entorno rico y con mucha información
 - Uso de información simbólica en el proceso de razonamiento
 - Uso de interfaces naturales en su comunicación con el usuario
 - Capacidad de aprendizaje como método de adaptación

Características de los SBC

En la construcción de sistemas con estas características:

- Debemos mantener el conocimiento del dominio y de resolución de problemas separado del mecanismo de control de la resolución
- Debemos incorporar conocimiento heurístico en la resolución (incompleto, aproximado, no sistemático).
- Debemos permitir una interacción estrecha con el usuario y/o entorno

Áreas de la IA involucradas en los SBC

Las características que buscamos en los SBC y el tipo de habilidades que necesitan hacen que se combinen diferentes áreas de la inteligencia artificial en su diseño y construcción, entre ellas:

- Representación del conocimiento (conocimiento sobre dominio, sobre la resolución, heurísticas, ...)
- Razonamiento e inferencia (lógica clásica, incompletitud, incertidumbre, tiempo, ...)
- Búsqueda heurística/Resolución de problemas
- Tratamiento del lenguaje natural (interfaces)
- Aprendizaje automático (adquisición del conocimiento del dominio, adaptación, ...)

Necesidad de los SBC

- Disponer del conocimiento de expertos altamente cualificados
- Poder ayudar/formar expertos/no expertos
- Preservar el conocimiento de expertos
- Obtener soluciones rápidas y justificadas
- Tratar grandes volúmenes de información
- Tener sistemas que tomen decisiones autónomas

Problemas resolubles mediante SBC

- Han de tener una complejidad suficiente que justifique el coste de su desarrollo
- Han de estar correctamente dimensionados
- Hemos de poder disponer del conocimiento experto necesario
- Hemos de poder plantear el problema como un proceso de razonamiento
- Han de poder estructurarse adecuadamente
- No tienen una solución mediante técnicas tradicionales
- Tenemos expertos cooperativos

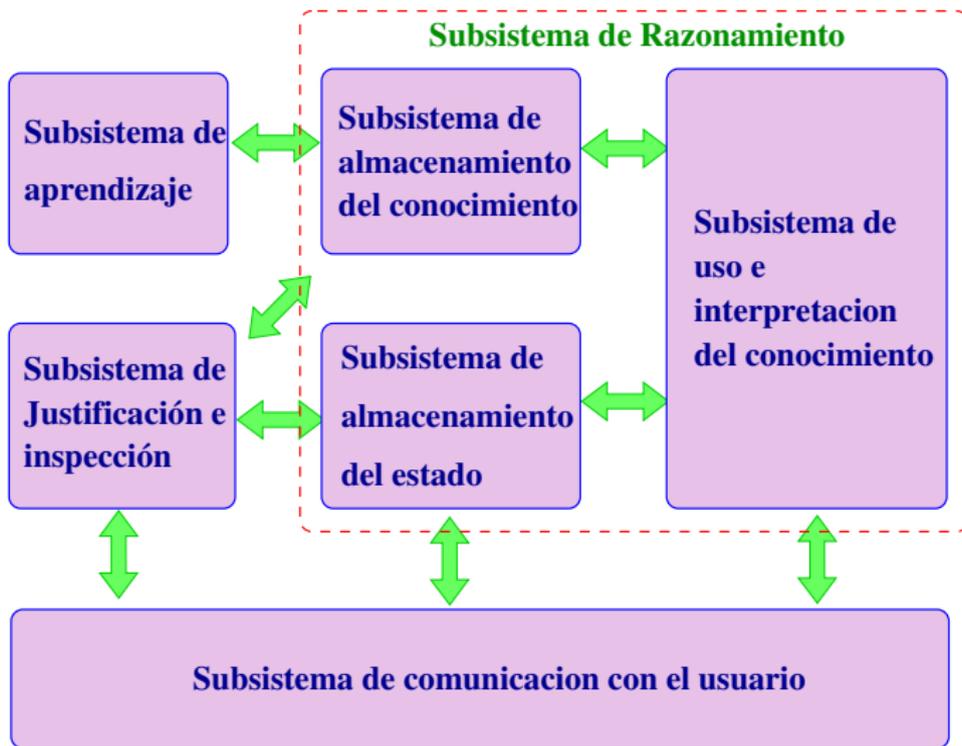
Problemas de los SBC

- Fragilidad
- Dificultad del control del razonamiento
- Poca reusabilidad del conocimiento o las estrategias de resolución
- Difícil integración del aprendizaje en el sistema
- Dificultad de adquirir el conocimiento a partir de expertos
- Dificultad en la validación de la correctitud/completitud del sistema

Áreas de aplicación de los SBC

- Se pueden encontrar en cualquier dominio en el que se necesite un conocimiento especializado
- Existen aplicaciones en multitud de dominios (medicina, ingeniería, predicción meteorológica, banca, ...)
- Problemas que involucran el análisis de un conjunto de evidencias (interpretación, diagnóstico, supervisión, predicción, ...)
- Problemas que involucran la construcción de una solución (diseño, planificación, configuración, ...)

Componentes de los SBC



SBC basados en sistemas de producción

- La resolución se obtiene a partir del proceso de razonamiento de un motor de inferencia
- El conocimiento del dominio está expresado mediante una ontología
- El conocimiento de resolución de problemas está almacenado habitualmente como reglas de producción o un formalismo equivalente

Almacenamiento del conocimiento

- Almacenará todo el conocimiento para resolver problemas en el dominio de aplicación
- Encontraremos tres tipos de conocimiento:
 - Conocimiento factual (objetos del dominio y sus características)
 - Conocimiento relacional (relaciones entre los objetos del dominio)
 - Conocimiento condicional (conocimiento deductivo sobre el problema)
- Los dos primeros conocimientos están descritos mediante la ontología de dominio
- El tercer conocimiento describirá el conocimiento relacionado con la resolución

Almacenamiento del conocimiento: Reglas

- El conocimiento condicional incluye:
 - Conocimiento deductivo (estructural): Describe los procesos de resolución de problemas como cadenas de deducción
 - Conocimiento sobre objetivos (estratégico): Orienta el proceso de resolución
 - Conocimiento causal (de soporte): Apoya al proceso de explicación de la resolución
- **Módulos de reglas**
 - Permite facilitar el desarrollo y el mantenimiento del sistema
 - Permite aumentar la eficiencia del proceso de razonamiento
 - Permite implementar estrategias de uso del conocimiento (meta-conocimiento, meta-reglas)

Almacenamiento del conocimiento: Meta-Reglas

- Describen conocimiento a alto nivel sobre la resolución del problema
- Permiten dirigir el control de la resolución
 - Activar y desactivar reglas/módulos
 - Decidir el orden de ejecución de reglas/módulos
 - Decidir estrategias de resolución, tratamiento de excepciones, incertidumbre, ...
- Son más difíciles de obtener de los expertos

Uso e interpretación del conocimiento

- Es habitualmente un motor de inferencia
- Aplicará su ciclo de ejecución para resolver el problema
 - Detección de reglas aplicables
 - Selección de la mejor regla (estrategia general o guiada por el metaconocimiento)
 - Aplicación de la regla

Almacenamiento del estado

- Guarda los datos iniciales del problema y los hechos obtenidos durante el proceso de resolución
- Puede guardar otro tipo de información necesaria para el control de la resolución y otros subsistemas
 - Orden de deducción de los hechos
 - Preferencias sobre el uso de los hechos
 - Reglas que generaron los hechos
 - Reglas activadas recientemente
 - Puntos de backtracking
 - ...

Justificación de la solución

- La posibilidad de justificar las decisiones da credibilidad al sistema
- También permite detectar deducciones erróneas
- Un sistema debería poder contestar **Porqué** y **Cómo**
- Diferentes niveles de justificación:
 - **Muestra:** Traza de los pasos de resolución
 - **Justificación:** Razones de los elementos que aparecen en la traza de la resolución (línea de razonamiento, preguntas, hechos, preferencias, subproblemas, ...)

Aprendizaje

- Por lo general el conjunto de problemas que se resuelven está acotado
- En algunos dominios es necesario adaptarse al entorno y resolver nuevos problemas
- El aprendizaje puede suceder:
 - Durante el proceso de construcción del SBC: Se substituye o complementa el proceso de adquisición con métodos de aprendizaje inductivo, se construye un modelo a partir de ejemplos
 - Durante el proceso de resolución: Se detectan y corrigen las resoluciones erróneas, se aprenden reglas de control que mejoran la eficiencia del proceso de resolución

Razonamiento basado en casos

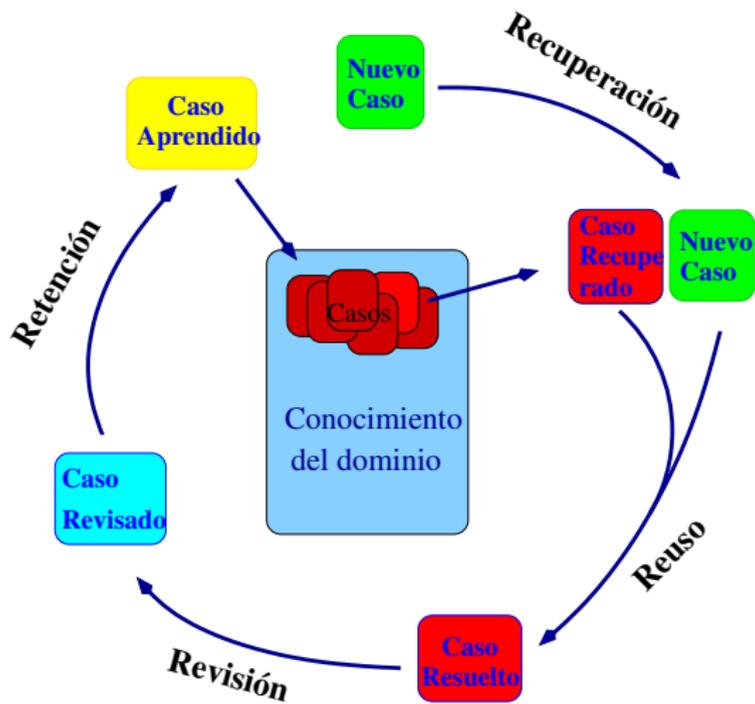
- La resolución de un problema se obtiene identificando una solución anterior similar
- Ventajas:
 - Reducen el problema de extracción del conocimiento
 - Facilitan el mantenimiento/corrección/extensión del sistema
 - Permite una resolución más eficiente
 - Permite explicaciones mas cercanas a la experiencia del usuario

Ciclo de ejecución

Consta de cuatro fases

- 1 **Recuperación:** Búsqueda de los casos almacenados más similares
- 2 **Reuso:** Obtenemos la solución del caso recuperado
- 3 **Revisión:** Evaluamos y adaptamos la solución recuperada
- 4 **Retención:** Comprobamos si es interesante guardar el caso

Ciclo de ejecución



Almacenamiento del conocimiento

- El conocimiento estará formado por **casos**
- Un caso es una estructura compleja (características, solución)
- Se almacenarán en la **base de casos** (estructura, indexación)
- Tendremos también conocimiento para:
 - Evaluar la similaridad entre los casos
 - Combinar/Adaptar las soluciones recuperadas
 - Evaluar las soluciones

Uso e interpretación del conocimiento

- Se basa en el ciclo de ejecución de razonamiento basado en casos
 - Búsqueda en la base de casos de los casos más similares
 - Recuperación de las soluciones de los casos
 - Combinación/adaptación de soluciones (procedimientos/razonamiento)

Almacenamiento del estado

- Información del caso actual
- Cálculo de los casos más similares
- Razonamiento para la evaluación/combinación/adaptación de las soluciones

Justificación - Aprendizaje

• Justificación

- Es parte de la información de los casos
- Se complementará con el razonamiento sobre la combinación/adaptación de las soluciones

• Aprendizaje

- Añadir nuevos casos (mas sencillo que en los sistemas de reglas)
- La solución debe ser suficientemente diferente (evaluación)
- Podemos olvidar casos (poco usados, parecidos a otros)

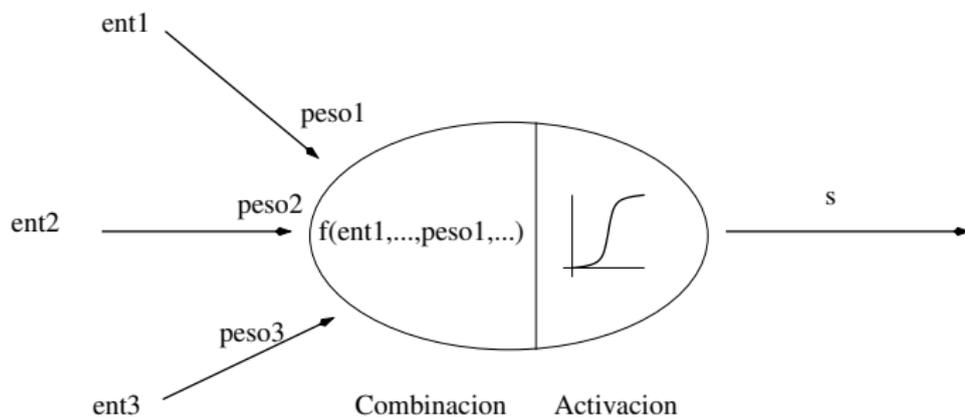
Otras metodologías

- Sistemas basados en redes neuronales
- Razonamiento basado en modelos
- Agentes Inteligentes/Sistemas Multiagente

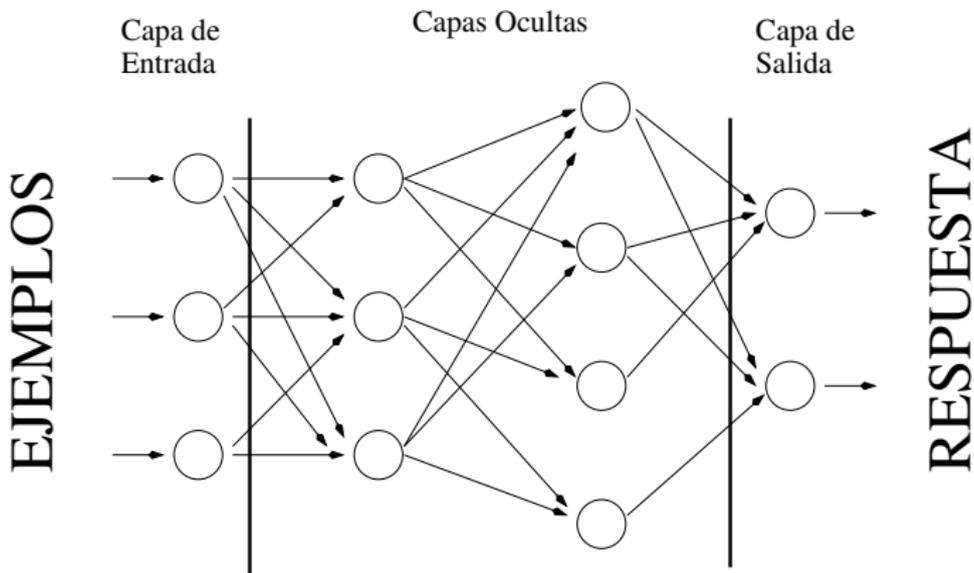
Redes neuronales

- Dentro del área de la Inteligencia Artificial conexionista
- El elemento base es la neurona (elemento de cómputo)
- **Neurona:** Entradas, salidas, estado, funciones para la combinación de las entradas y el estado y función para generar la salida
- Las neuronas se organizan en redes con diferentes capas
- La red asocia unas entradas (datos del problema) a unas salidas (solución del problema)
- La red se debe entrenar (ejemplos resueltos) para que aprenda a resolver el problema (asociación)

Redes neuronales



Redes neuronales



Razonamiento basado en modelos

- Construimos un modelo del comportamiento del sistema
- Este modelo se basa en información cualitativa
- Razonando sobre el modelo podemos predecir las consecuencias de nuestras acciones
- Utilizamos razonamiento de sentido común en la resolución
- *Física naïf*

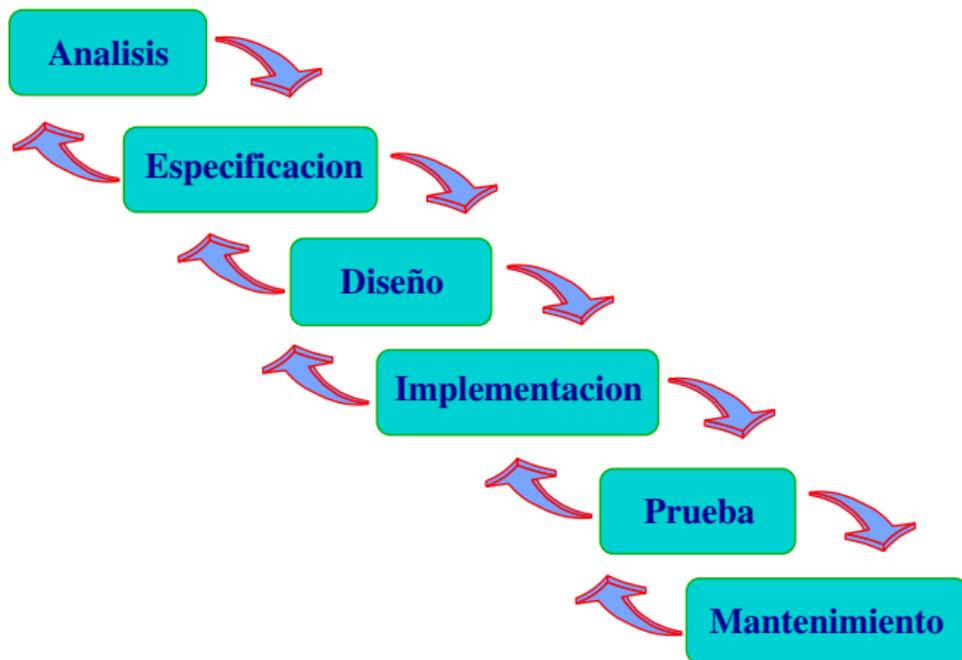
Agentes inteligentes/Sistemas multiagente

- Nos alejamos de una visión monolítica de los sistemas inteligentes
- Un agente inteligente resuelve una tarea “sencilla”
- El problema global se resuelve en cooperación/coordiación
- Otras áreas involucradas (organización, cooperación, coordinación, negociación, división del trabajo, comunicación, razonamiento sobre otros, ...)
- Ventajas: Sistemas mas flexibles, reconfiguración/reorganización para otras tareas (componentes) \implies resolver más problemas
- Relacionado con computación Grid y servicios web

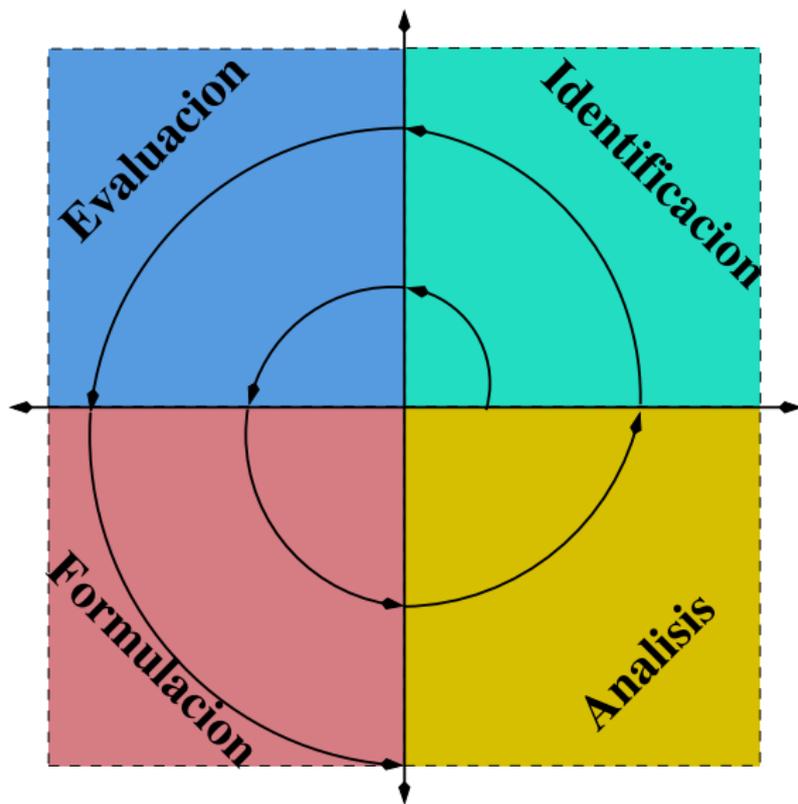
Desarrollo de SBC

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

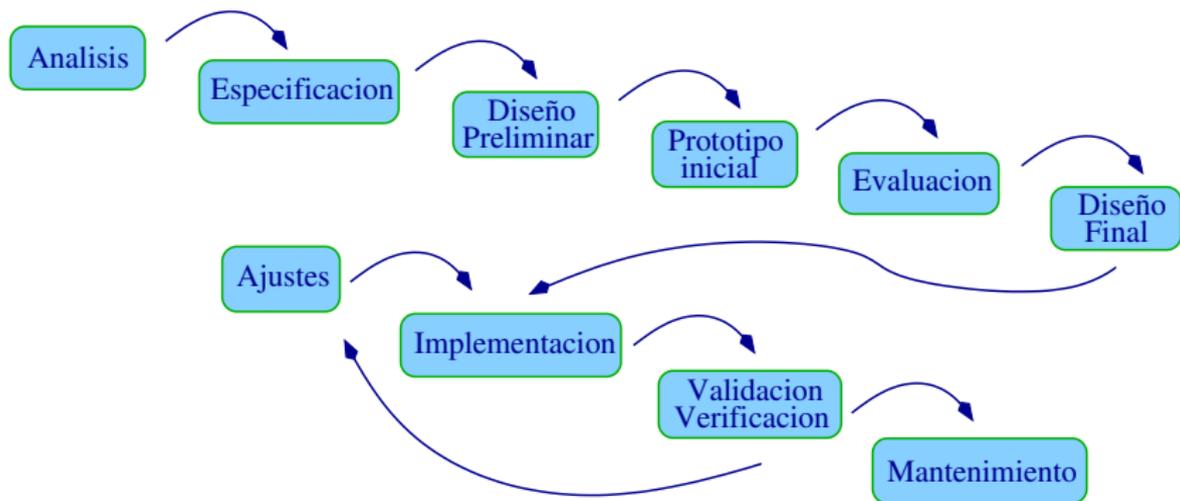
Diferencias de los SBC

- 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/ complejidad 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