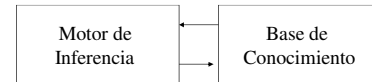


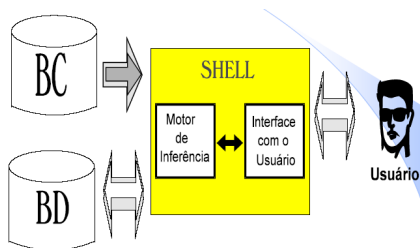
Sistema Basado en Conocimiento

Construyendo un SBC

- Se pueden combinar estos 2 elementos de la siguiente forma



Estructura Básica de SBC



SBC: Base de Conocimiento

- Es un tipo especial de base de datos para la gestión del conocimiento.
- Provee los medios para la recolección, organización y recuperación computarizada de conocimiento.
- La base de conocimiento está compuesta de hechos y reglas que el sistema conoce sobre el dominio del problema

Componentes Básicos de SBC

- Sistemas basados en conocimientos
 - Conocimiento + inferencia
- SBCs tienen 2 componentes esenciales
 - Una base de conocimiento que captura el conocimiento específico del dominio
 - Un motor de inferencia el cual consiste de algoritmos que manipulan conocimiento representado en la base de conocimiento.

SBC: Base de Conocimiento

- El conocimiento puede ser representado usando técnicas como:
 - Reglas de producción
 - Redes semánticas
 - Frames
 - Scripts, etc
- También pueden ser una mezcla de diferentes métodos que forman parte de sistemas híbridos

Redes Semánticas

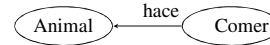
- Redes Semánticas son una tentativa de formalizarse como nuestro conocimiento es organizado en la memoria.
- Redes Semánticas son compuestas de *nodos* y *links* etiquetados.
- Cada nodo representa un objeto o propiedad de un objeto.
- Cada link representa la relación entre dos nodos.

RC - Redes Semánticas

sor

Ej: Red Semántica Simple (cont.)

↪ La sentencia "Animales comen" puede ser representada por la siguiente red:



- "Animal" y "Comer" son representados por nodos.
- La relación entre ellos (este animal come) es representado por el link rotulado "hace".
- Simplemente, se puede leer como "Animal hace Comer".

RC - Redes Semánticas

sor

Historia de las Redes Semánticas

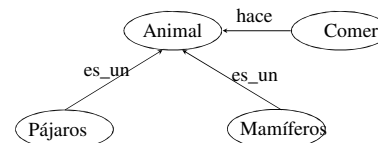
- Originalmente a idea de redes semánticas fue propuesta en 1913 por Selz como una explicación de fenómenos psicológicos.
- En 1966, Quillian implemento aquellas ideas y mostró como el significado podría ser representado como la relación entre dos objetos.
- Representaciones mas complicadas tales como frames son realces de esta idea.

RC - Redes Semánticas

sor

Red Semántica Simple (cont.)

- "Mamíferos y Pájaros son animales" puede, ahora, ser aumentada usándose el link "es_un":



↪ Se Puede leer esta nueva sentencia como:
 "Pájaro es un Animal" y
 "Mamífero es un Animal".

RC - Redes Semánticas

sor

Ej: Red Semántica Simple

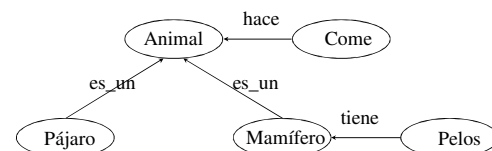
- Redes Semánticas explicitan la *relación* entre objetos y propiedades.
- Por ejemplo, considere **algunas** cosas que sabemos sobre animales:
 - Animales comen;
 - Mamíferos y pájaros son animales;
 - Mamíferos tiene pelos; y
 - Perros son mamíferos

RC - Redes Semánticas

sor

Una Red Semántica Simple (cont.)

- También se puede aumentar a la red la sentencia "Mamíferos tienen pelos" :

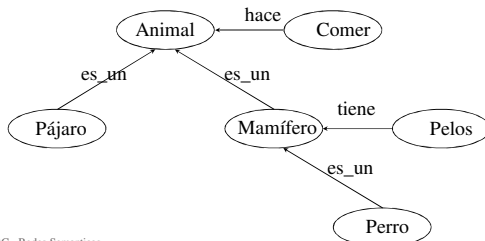


RC - Redes Semánticas

sor

Una Red Semántica Simple (cont)

- Y, por último, se puede aumentar "Perros son mamíferos":



RC - Redes Semánticas

sor

Búsqueda como una herramienta Exploratoria

- Podemos suponer que lo perros comen, y usar búsqueda sobre la red para explicar esto (se el puede).
- Buscando a partir del nodo "Perro", podemos decir que "Perro es un Mamífero", "Mamífero es un Animal" y "Animal hace Comer".
Esto es **una** explicación para "Perros comen".

RC - Redes Semánticas

sor

Transitividad en Redes Semánticas

- Redes Semánticas son naturalmente transitivas.
- Podemos concluir de la red desarrollada que si "Perro es un Mamífero" y "Mamífero es un Animal" entonces "Perro es un Animal".
- Entretanto, **no** es posible concluir que:
 - "Perro es un Pájaro"; o
 - "Pájaro tiene pelos".

RC - Redes Semánticas

sor

Busca exhaustiva de información

- Se quisiéramos encontrar todo lo que podemos aprender sobre perros, solamente necesitamos usar **Busca en Profundidad** a partir de "Perro".
- De esa manera, podríamos encontrar que "perro son mamíferos", "perros tienen pelos", "perros son animales" y "perros comen".

RC - Redes Semánticas

sor

Busca en Redes Semánticas

- La Búsqueda en Redes Semánticas puede ser usada de varias maneras para extraer informaciones.
- Por ejemplo, la búsqueda puede ser usada:
 - como una herramienta explicativa;
 - para explorar exhaustivamente un tópico; y
 - para encontrar la relación entre dos objetos.

RC - Redes Semánticas

sor

Intersección de Búsqueda

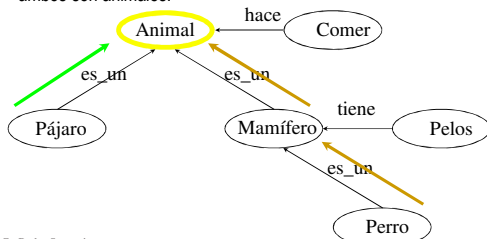
- Se quisiéramos encontrar si "perros" y "Pájaros" están relacionados, entonces podemos ejecutar, a partir de ambos nodos, una búsqueda en largura.
- La intersección nos da una pista sobre la relación entre los nodos.
- Esto es llamado activación distribuida o intersección de búsqueda.

RC - Redes Semánticas

sor

Intersección de Búsqueda (cont)

- Partiendo de "Perro" e "Pájaro" podemos encontrar que ambos son animales:



RC - Redes Semánticas

SOF

El Frame "Perro"

- El frame "Perro" podría ser expandido aumentándose nuevos slots y valores para el frame:

Slots	Perro	Es_un	Mamífero	Valores
		nombre		
		raza	Default: pequines	
		Pelo	Default: Largo	
		Sexo	Macho o Hembra	

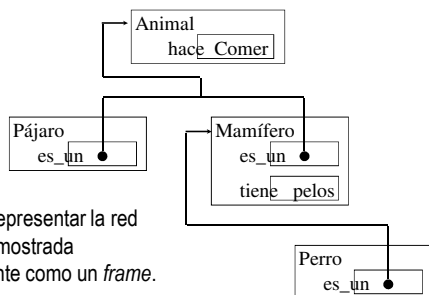
Frames

- Frames (Minsky 1975) son mas poderosos que las redes semánticas, porque:
 - Ellos proveen una representación mas estructurada que la red semántica;
 - Tanto información como relación pueden ser especificados en un frame;
 - Ellos también pueden contener procedimientos.
- Frames pueden ser representados de una forma gráfica similar a redes semánticas.

Aspectos Generales de un Frame

- Slots son atributos del frame que pueden tener valores particulares, llamados valores.
- Valores pueden ser un valor absoluto, un intervalo o un valor default.
- Un frame genérico, tal como el frame "Perro", es una clase frame.
- Una instancia de una clase frame es simplemente un frame con valores específicos, así como Rastus, o Perro, es una instancia de la clase de perros.

Redes Semánticas como Frames



- Podemos representar la red semántica mostrada anteriormente como un frame.

Una Instancia del Frame "Perro"

- "Rastus" - Una instancia de la clase "Perro":

Perro	Es un	Mamífero
	nombre	Rastus
	raza	German Shepherd
	Pelo	Longo
	Sexo	Macho

Frames y Demons

- Procedimientos que están dentro de *frames* son llamadas *demons*.
- Un ejemplo de un *demon* es un procedimiento para calcular a área de un cuadrado dado el tamaño de un de los lados.
- Así el valor del área no precisa estar representado y si puede ser calculado a partir de otras informaciones en la instanciación del *frame*.

Consideraciones Finales

- Objetos en la Programación Orientada a Objetos son *muy* similares a los *frames*.
- Por esa razón, Lenguajes OO son una buena opción para la implementación de sistemas de *frames*.
- Vimos anteriormente que los procedimientos pueden ser alojadas dentro de *frames* para aumentar su flexibilidad y performance.

El Frame "Cuadrado"

Cuadrado	
Tam. del lado	<input type="text"/>
Área	<input type="text"/>

⚡ La clase *frame*, para cuadrado, tiene un demon en Área que llena el valor en Tam. del lado.

- Cuando el lo encuentra, el calcula a área del cuadrado.

Cuadrado	
Tam. del lado	<input type="text" value="5"/>
Área	<input type="text" value="25"/>

Historia de Scripts

- *Scripts* (Schank e Abelson 1977) son una especialización de *frames* proyectados para manipular situaciones además de objetos.
- En una red semántica o en *frames*, nodos son objetos, y los links entre los objetos representan una gama de relaciones.
- En *scripts*, los nodos son *eventos*, y los links entre ellos son simplemente causales:
 - Esto es, un evento provoca el próximo.

Frames y herencia

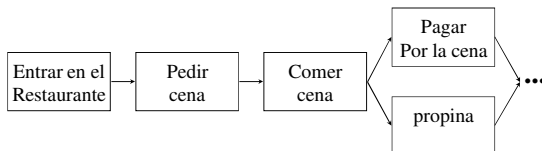
- En el ejemplo animal/mamífero/perro, el nivel mas bajo *hereda* las propiedades de los niveles superiores.
- Por ejemplo: Perro tiene pelos, pues ellos son mamíferos y mamíferos tienen pelos.
- Herencia es una característica poderosa de *frames*, porque la información puede ser especificada en un nivel mas genérico, evitándose, así, redundancia.

Construyendo un Script

- Un *Script* es como un *script* cinematográfico.
- Como en un *script* de cine necesitamos considerar el número de elementos cuando proyectamos el *script*:
 - Cuales son los *papeles* de los objetos/personas en el *script*;
 - Cuales *objetos de la escena* se relaciona al *script*;
 - Cuales son las motivaciones o *entradas condicionales* para ejecutar el *script*;
 - Cuales escenas están por ocurrir; y
 - En cual *orden* ellas deben ocurrir; etc.

Un Script Básico

- Antes de proyectamos el *script*, necesitamos de una secuencia básica inicial.
- Por ejemplo, en la ida a un restaurante hay una secuencia de eventos que podemos esperar:



E Script Restaurante (cont)

Escena 1: Entrar

- Estacionar el carro
- Entrar en el Restaurante
- Esperar por una Mesa
- o
- Ir hasta la Mesa
- Leer el Menú

Escena 2: Pedir la cena

Corte de un Script

- Es posible cortar cada uno de los eventos en una serie de sub-eventos. Por ejemplo con relación al evento entrar en el restaurante, se puede esperar:



Scripts

- *Scripts* pueden ser usados para:
 - Contar historias sobre una secuencia de eventos;
 - Responder preguntas tales como "lo que sucede si la carne del cliente estuviese quemada?"
 - Y para seguir a los eventos para que nos lleven a alguna decisión.
- *Scripts* son muy similares a los *frames*, son codificados de la misma forma y son, normalmente, considerados como una sub-clase de *frames*.

El Script Restaurante

- Colocando los eventos junto con a los demás elementos, podríamos imaginar el *script* "Restaurant" asignando apenas algunas cosas, tales como:

SCRIPT RESTAURANTE

Papeles: Cliente, mozo, cocina.

Objetos de cena: Mesas, sillas, tenedores, cuchillo, platos, vasos, botellas de vino...

Entradas Condicionales: Cliente está Hambriento; Cliente está vestido in apropiadamente; Cliente tiene dinero.

Motor de Inferencia

- El motor de inferencia incluye un interpretador de reglas de una base de conocimiento y ejecuta ítem de una agenda
- Incluye también:
 - Un agendador
 - Un verificador de consistencia de las soluciones que surge

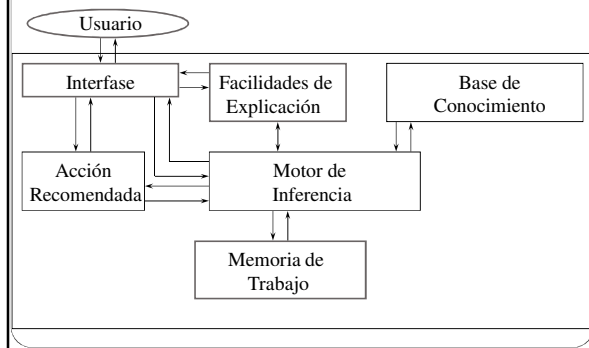
Componentes Adicionales

- Teniendo el núcleo de un SBC, que otros componentes son necesarios?
 - Un usuario;
 - Un modulo de usuarios interactúan con el sistema;
 - Un lugar para almacenar el conocimiento usado no trabajo;
 - Una manera de conseguir ayuda del sistema.
- Eso es necesario para interactuar con el motor de inferencia y la base de conocimiento.

Interfase

- La interfase es un procesador de lenguaje orientado a procesar y producir comunicación entre usuario y sistema.
- Esto generalmente ocurre en lenguaje natural, siendo complementada por menús gráficos
- El procesamiento de lenguaje natural es por lo tanto importante en el SBC

Una estructura completa de SBC



Memoria de Trabajo

- La memoria de trabajo del sistema almacena condiciones iniciales hipótesis intermedias, decisiones y soluciones finales.
- Información e clasificación en 3 tipos
 - Planos (Como resolver el problema)
 - Agendas (acciones potenciales a ser ejecutadas)
 - Soluciones (soluciones candidatas y soluciones intermediarias)

Usuario

- Los SBC están diseñados para interactuar con varios usuarios actuando de diferentes maneras y circunstancias:
 - Modo consultor
 - Modo instructor
 - Modo acompañante
 - Modo cooperativo

Facilidades de Explicación

- El modulo que facilita la explicación puede justificar las conclusiones, y auxiliar a explicar el comportamiento del SBC
- Esto es hecho a través de preguntas iterativas:
 - Porque
 - Como

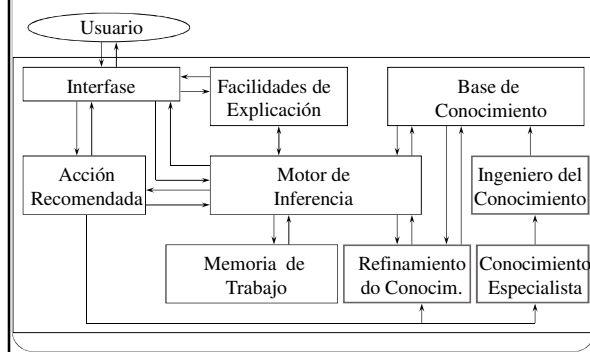
Adquisición del conocimiento

- Como el sistema adquiere conocimiento
 - Como el conocimiento es conseguido inicialmente
- El sistema adquiere nuevos conocimientos
- Como este conocimiento puede ser probado
- Que componentes son necesarios para aumentar esos conocimientos

El Ingeniero del conocimiento

- El equipo de ingenieros del conocimiento realiza las siguientes tareas
 - Estructura el área del problema
 - Interpreta, traduce e integra conocimiento especialista al sistema.
 - Traza analogías.
 - Presenta contra ejemplos.
 - Chequea la consistencia del conocimiento

Estructura para SBC Completo



Refinamiento del conocimiento

- Futuros SBC podrán ser capaces de analizar aprender y mejorar sus propias performances resultando una base de conocimiento mas apropiada y un raciocinio mas efectivo
- Mientras tanto, en los SBCs actuales esta tarea la realiza el ingeniero en conocimiento

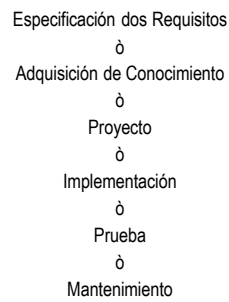
Conocimiento experto

- Adquirir conocimiento especialista para una base de un SBC incluye:
 - Obtener información de los especialistas y/o fuentes documentadas
 - Clasificación de esta información en declarativa o procedimental
 - Codificación de esta información en un formato usado por el SBC
 - Chequear la consistencia del conocimiento

Shell de un SBC

- El shell de un SBC es un SBC con una base vacía.
- El Shell esta destinado a que los no programadores usufructúen los esfuerzos de los programadores que ya resolvieron un problema similar

Proceso de Desarrollo de SBC



Ejemplo Sistema Basado en Conocimiento

- Los sistemas expertos basados en reglas
 - Constituyen la más sencilla de las metodologías utilizadas en sistemas expertos.
 - Trabajan mediante la aplicación de reglas, comparación de resultados y aplicación de nuevas reglas basadas en una situación modificada.